The rules used in the design of the valve controller and its logic were:

1. Any valve should be available to be closed using the switches at any time regardless of interlock or key state.
2. The isolation valve for the turbo pump must close immediately after the turbo trips, but there will be a time delay of up to 5 minutes after the pump is tripped before the vent valve is opened by the controller.
3. There will be a solenoid feeding N2 to the vent valve which will open when the turbo is NOT Up-To-Speed. There must also be a manual valve in parallel with the solenoid so N2 can be fed to the vent valve in the case of a planned vent.
4. The cryopump gate valve must close if the compressor stops. The compressor will have an interlock on its power which closes the gate valve.
5. In the case of a power outage, all valves will close.
6. The cryopump and turbo gate valves can only be opened with the bypass keys engaged.
7. If either ion pump loses power, its gate valve can stay open since the chamber will stay under vacuum anyway.
8. The RESET button will reset all open gate valves to the closed state and RESET the program in the PLC.

The general layout of the system is:

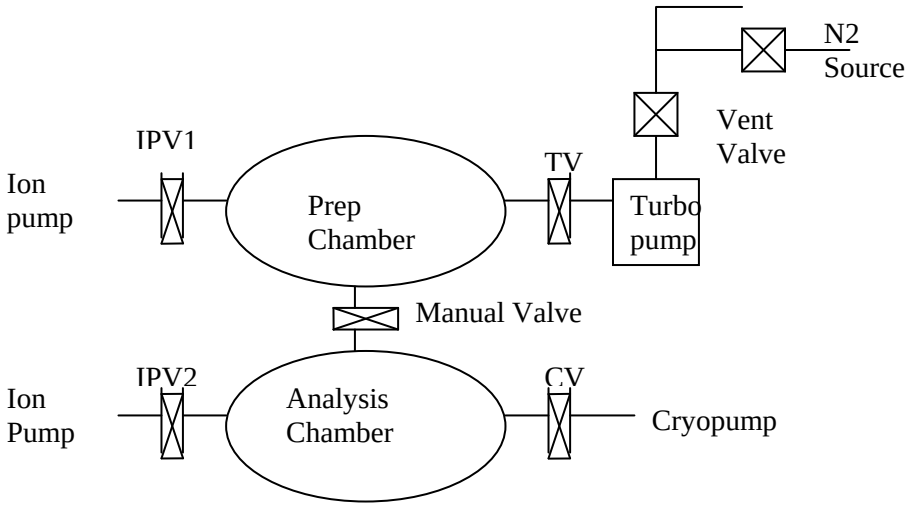


Figure 1 Cartoon of SES 4000 System

The rules for valve operation are:

VALVE	INTERLOCKS
IPV1	Switch open/close control, Close on power loss
IPV2	Switch open/close control, Close on power loss
CV	Close on compressor, power loss or switch, only open with key
TV	Key enable to open. Close with switch, power loss or key removal.
N_2 Valve	Open when turbo not up to speed

The controller is shown in Figure 2. The Red button protected by the copper pipe is the RESET button. The gate valve control switches are located in a remote head mounted to the Scienta frame, shown in Figure 3.

Figure 2. SES 4000 valve controller mounted in frame

Figure 3. SES 4000 Valve control switches mounted under manipulator arm

To open a valve, the conditions described in the Table of State (below) must be satisfied and then the small toggle switch on the valve control box shown in Figure 3, must be MOMENTARILY switched toward the left for Closed or the right for Open. If the operation is successful, the appropriate light will be illuminated. In Figure 3, for example, the cryopump valve and the ion pump 1 valve are Open. The turbo pump valve and the ion pump 2 valve are Closed.

Table of all states and actions.

	CRYO VALVE	TURBO VALVE	ION PUMP 1 VALVE	ION PUMP 2 VALVE	N2 VALVE
Power loss					
Ion Pump 1 Switch OPEN					
Ion Pump 1 Switch CLOSED					
Ion Pump 2 Switch OPEN					
Ion Pump 2 Switch CLOSED					
Cryopump Switch OPEN+ Cryopump Key Sw ENABLE					
Cryopump Switch CLOSED					
Cryo-compressor AC OFF					
Turbo Pump Switch OPEN					
Turbo Pump Switch CLOSED					
Turbo Pump Key Sw DISABLE					
Turbo Up-to-Speed NOT UP TO SPEED					
Turbo Up-to-Speed UP TO SPEED					
RESET Switch Engaged					

KEY:

OPEN CLOSED DON'T CARE

To determine the valve state from the table, look at the valves at the top and read down the columns. If any of the actions on the left are active, that row should be considered 'Active'. Finally, if any of the active rows has a red box in it, the valve will be closed. The valve will only open or stay open so long as all the active rows in the column are yellow or green.

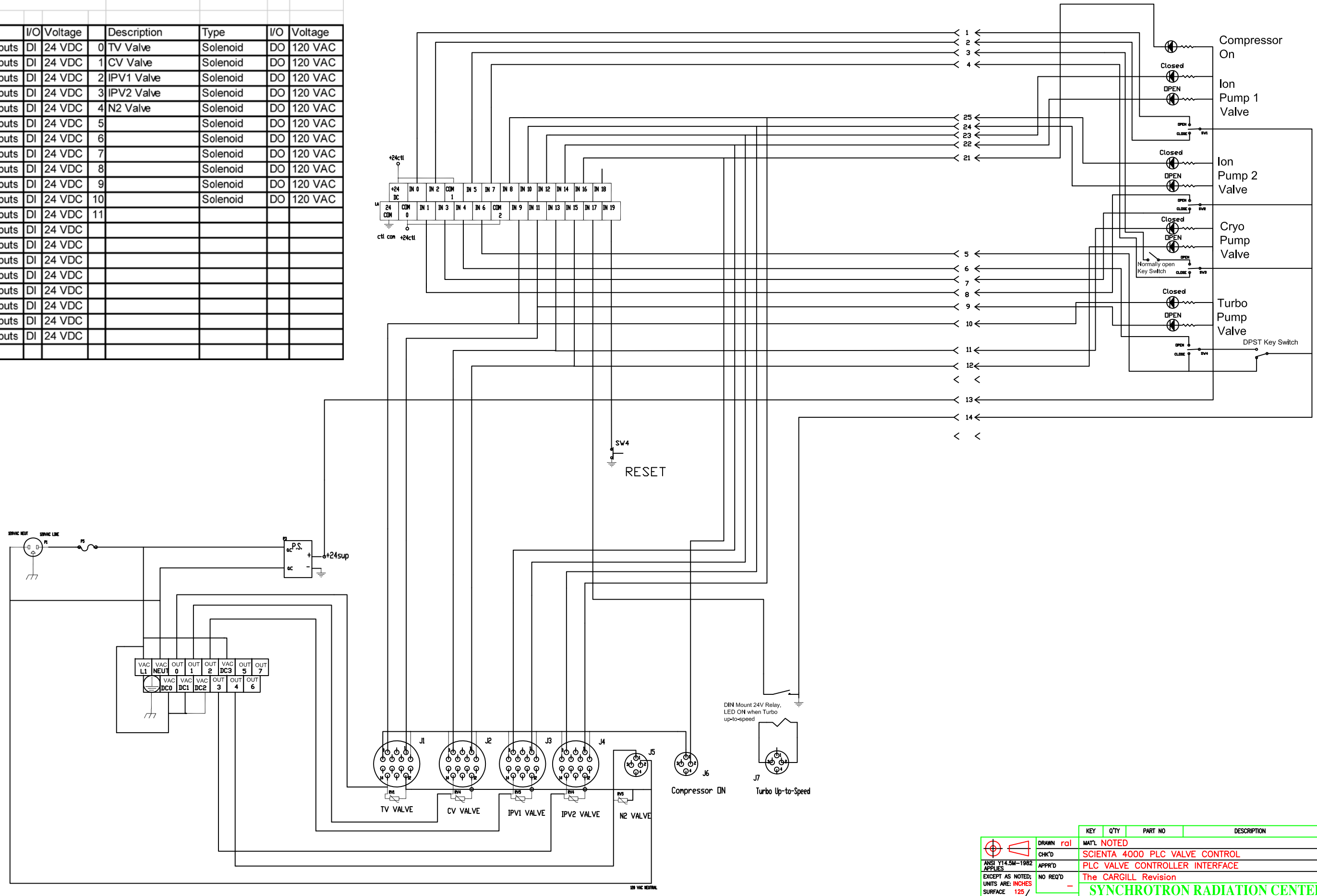
For example, if the Turbo pump gate valve switch is pushed to the OPEN position, but the Turbo pump key switch is not Enabled, the valve will NOT open.

INITIAL TURN-ON OR RECOVERY FROM EXTENDED POWER OUTAGE:

1. Verify that AC power light on the PLC is lit.
2. Push the RESET button. (The red button on the side of the chassis mounted at the bottom of the Scienta frame) This will shut all the valve except the N2 supply valve.
3. Check the chamber pressure gauges. You may have to re-start them. If the gauges will not re-start or the chamber pressure is too high to start the ion pumps, call Scienta contact person.
4. If the chamber pressure is low enough, open the ion pump gate valves by pushing the switches to the OPEN position and begin pumping out the chamber.

Base Unit - 1762-L32BWA

	Description	Type	I/O	Voltage		Description	Type	I/O	Voltage
0	IPV1 Open Cmd	Switch Inputs	DI	24 VDC	0	TV Valve	Solenoid	DO	120 VAC
1	IPV2 Open Cmd	Switch Inputs	DI	24 VDC	1	CV Valve	Solenoid	DO	120 VAC
2	IPV1 Closed Cmd	Switch Inputs	DI	24 VDC	2	IPV1 Valve	Solenoid	DO	120 VAC
3	IPV2 Closed Cmd	Switch Inputs	DI	24 VDC	3	IPV2 Valve	Solenoid	DO	120 VAC
4	TV Open Cmd	Switch Inputs	DI	24 VDC	4	N2 Valve	Solenoid	DO	120 VAC
5	CV Open Cmd	Switch Inputs	DI	24 VDC	5		Solenoid	DO	120 VAC
6	TV Closed Cmd	Switch Inputs	DI	24 VDC	6		Solenoid	DO	120 VAC
7	CV Closed Cmd	Switch Inputs	DI	24 VDC	7		Solenoid	DO	120 VAC
8	IPV2 Out Limit	Switch Inputs	DI	24 VDC	8		Solenoid	DO	120 VAC
9	TV In Limit	Switch Inputs	DI	24 VDC	9		Solenoid	DO	120 VAC
10	IPV2 In Limit	Switch Inputs	DI	24 VDC	10		Solenoid	DO	120 VAC
11	TV Out Limit	Switch Inputs	DI	24 VDC	11				
12	IPV1 Out limit	Switch Inputs	DI	24 VDC					
13	CV In Limit	Switch Inputs	DI	24 VDC					
14	IPV1 In Limit	Switch Inputs	DI	24 VDC					
15	CV Out Limit	Switch Inputs	DI	24 VDC					
16	Compressor On	Switch Inputs	DI	24 VDC					
17	Turbo Up-to-Speed	Switch Inputs	DI	24 VDC					
18		Switch Inputs	DI	24 VDC					
19	Reset Switch	Switch Inputs	DI	24 VDC					



CHANGE LETTER	CHGD BY	CHKD BY	DATE	CHANGE

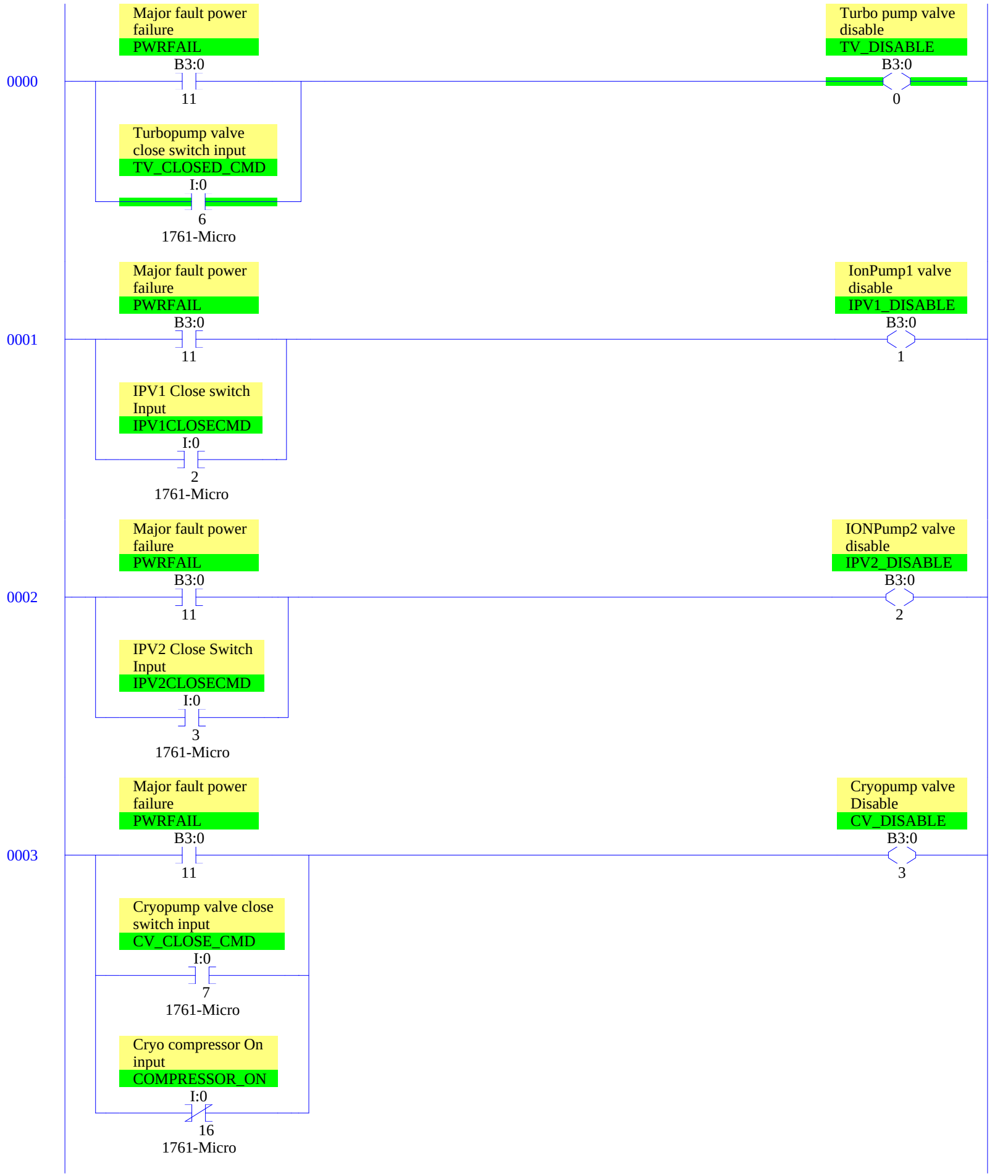
KEY	Q'TY	PART NO	DESCRIPTION
DRAWN	ral	MAT'L	NOTED
CHK'D			SCIENTA 4000 PLC VALVE CONTROL
APPR'D			PLC VALVE CONTROLLER INTERFACE
NO REQ'D			The CARGILL Revision

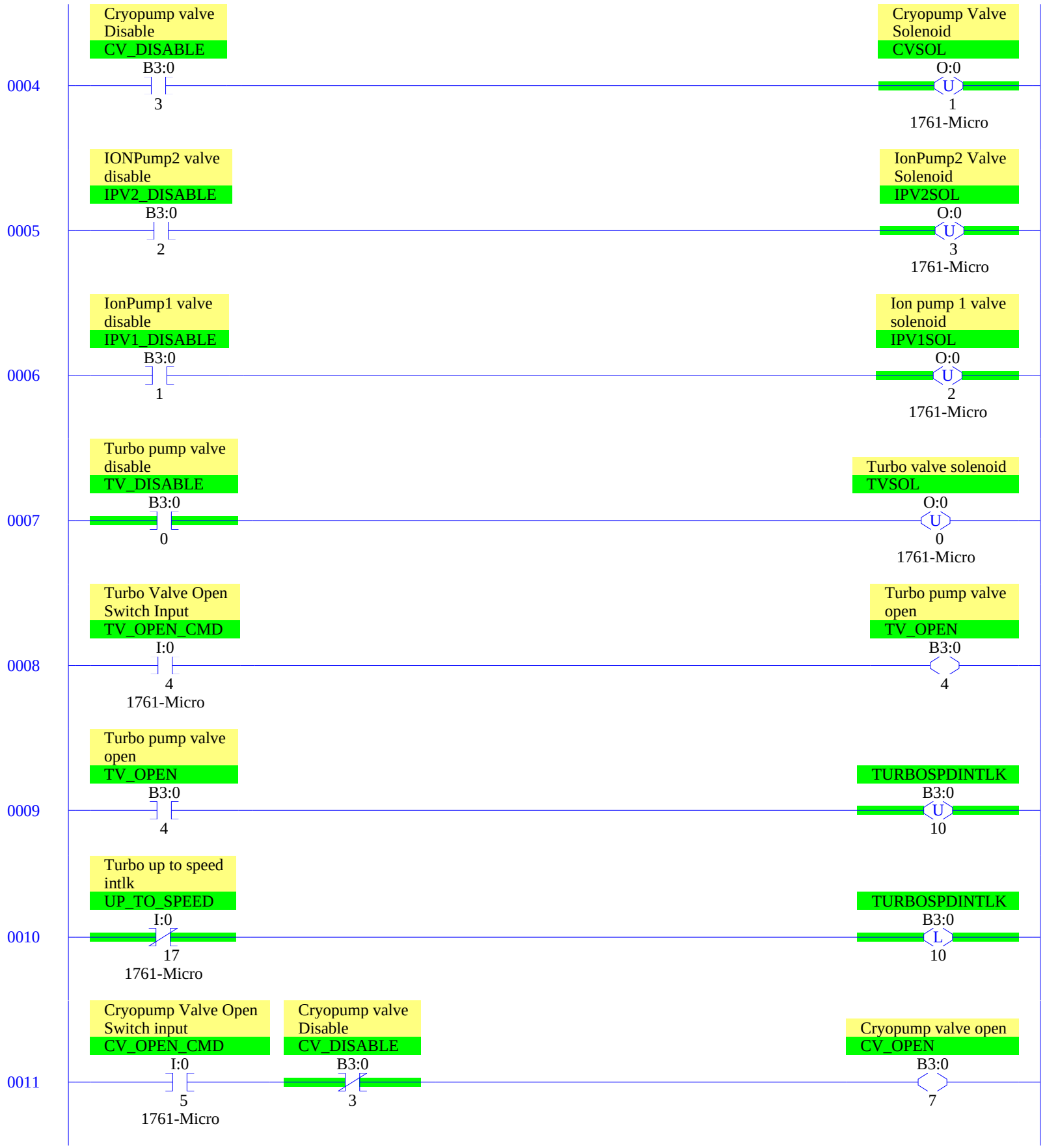
ANSI Y14.5M-1982 APPLIES.
EXCEPT AS NOTED; UNITS ARE: INCHES
SURFACE FINISH: 125
3 PLACE DECIMAL ± .005
2 PLACE DECIMAL ± .010
1 PL DEC., FRAC. ± .015 ± 1/64
ANGULAR TOL'S ± 0.5°

SYNCHROTRON RADIATION CENTER
UNIVERSITY OF WISCONSIN (608) 877-2000

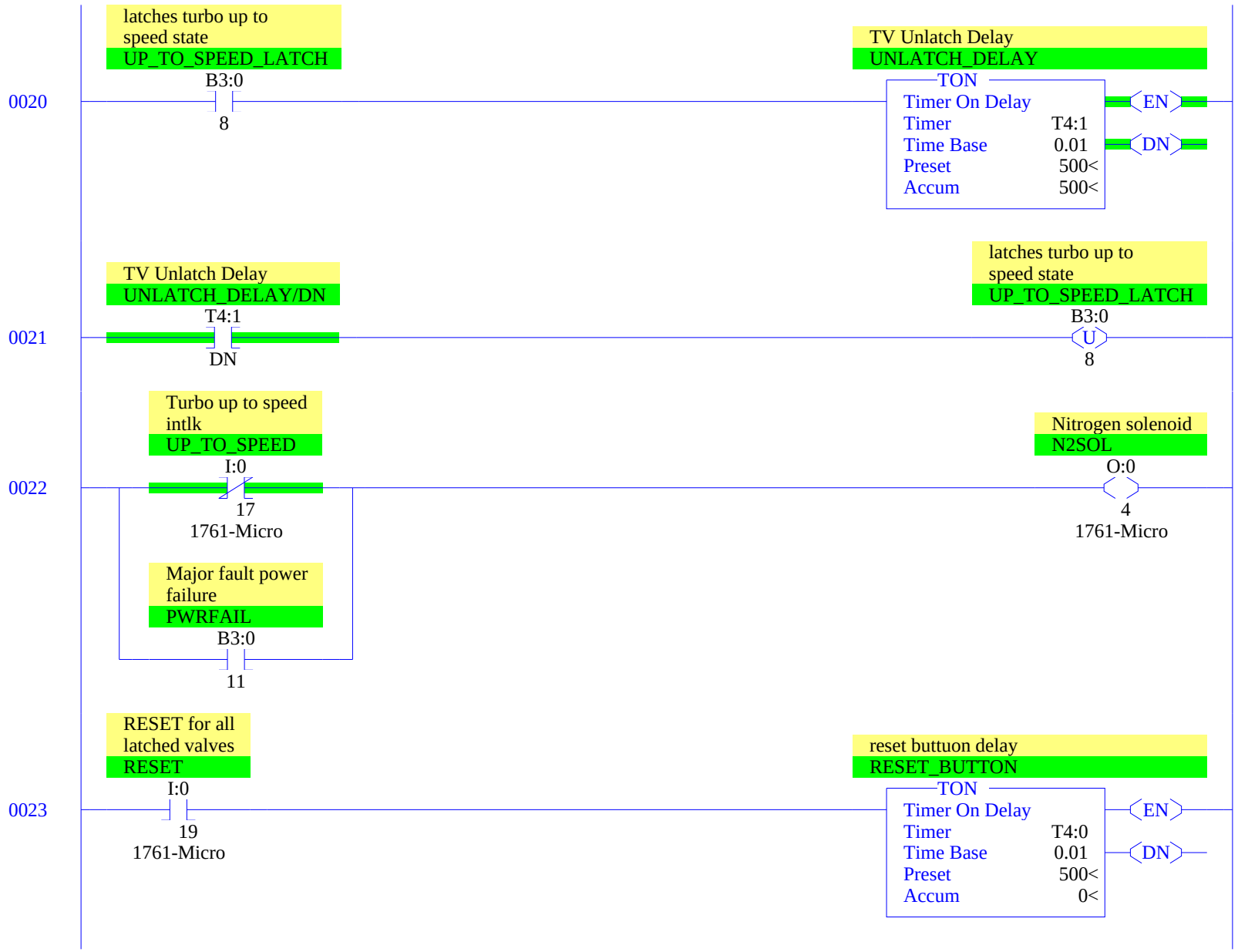
SCALE	USED ON	DWG NO	REV
FULL	0000E000	XXXXBXXX	—
DATE 06/27/06			

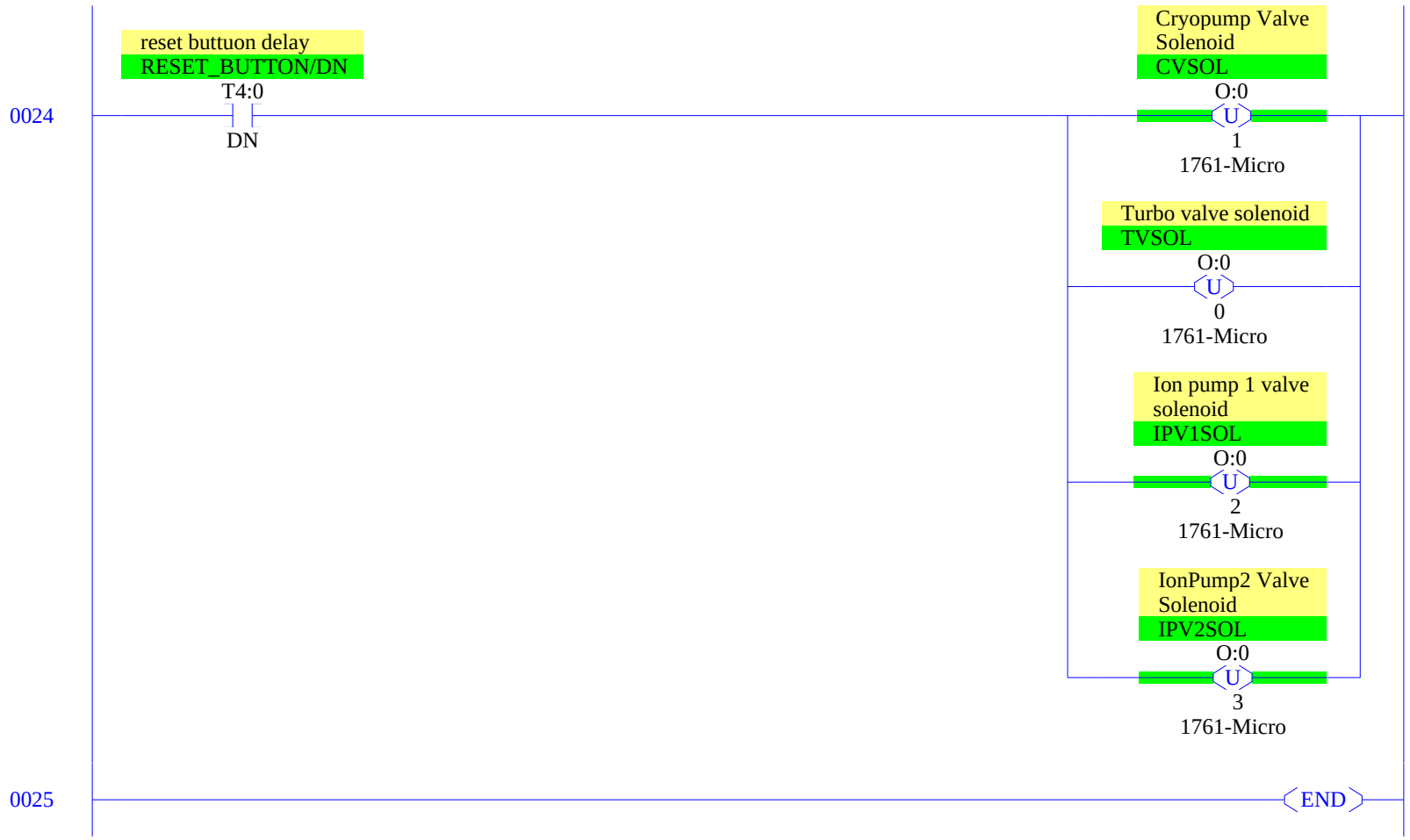
Drawn by AutoCAD R13


0000
Major fault power failure
PWRFAIL
B3:0
11
Turbopump valve close switch input
TV_CLOSED_CMD
I:0
6
1761-Micro
Turbo pump valve disable
TV_DISABLE
B3:0
0
0001
Major fault power failure
PWRFAIL
B3:0
11
IPV1 Close switch Input
IPV1CLOSECMD
I:0
2
1761-Micro
IonPump1 valve disable
IPV1_DISABLE
B3:0
1
0002
Major fault power failure
PWRFAIL
B3:0
11
IPV2 Close Switch Input
IPV2CLOSECMD
I:0
3
1761-Micro
IONPump2 valve disable
IPV2_DISABLE
B3:0
2
0003
Major fault power failure
PWRFAIL
B3:0
11
Cryopump valve close switch input
CV_CLOSE_CMD
I:0
7
1761-Micro
Cryo compressor On input
COMPRESSOR_ON
I:0
16
1761-Micro
Cryopump valve Disable
CV_DISABLE
B3:0
3


0004
Cryopump valve Disable
CV_DISABLE
B3:0
3
Cryopump Valve Solenoid
CVSOL
O:0
U
1
1761-Micro
0005
IONPump2 valve disable
IPV2_DISABLE
B3:0
2
IonPump2 Valve Solenoid
IPV2SOL
O:0
U
3
1761-Micro
0006
IonPump1 valve disable
IPV1_DISABLE
B3:0
1
Ion pump 1 valve solenoid
IPV1SOL
O:0
U
2
1761-Micro
0007
Turbo pump valve disable
TV_DISABLE
B3:0
0
Turbo valve solenoid
TVSOL
O:0
U
0
1761-Micro
0008
Turbo Valve Open Switch Input
TV_OPEN_CMD
I:0
4
1761-Micro
Turbo pump valve open
TV_OPEN
B3:0
4
0009
Turbo pump valve open
TV_OPEN
B3:0
4
TURBOSPDINTLK
B3:0
U
10
0010
Turbo up to speed intlk
UP_TO_SPEED
I:0
17
1761-Micro
TURBOSPDINTLK
B3:0
L
10
0011
Cryopump Valve Open Switch input
CV_OPEN_CMD
I:0
5
1761-Micro
Cryopump valve Disable
CV_DISABLE
B3:0
3
Cryopump valve open
CV_OPEN
B3:0
7

0012
IPV1 Open Switch Input
IPV1OPENCMD
I:0
0
1761-Micro
IonPump1 valve disable
IPV1_DISABLE
B3:0
1
IonPump1 valve open
IPV1_OPEN
B3:0
5
0013
IPV2 Open Switch input
IPV2OPENCMD
I:0
1
1761-Micro
IONPump2 valve disable
IPV2_DISABLE
B3:0
2
IonPump2 valve open
IPV2_OPEN
B3:0
6
0014
Turbo pump valve open
TV_OPEN
B3:0
4
Turbo valve solenoid
TVSOL
O:0
L
0
1761-Micro
0015
Cryopump valve open
CV_OPEN
B3:0
7
Cryopump Valve Solenoid
CVSOL
O:0
L
1
1761-Micro
0016
IonPump1 valve open
IPV1_OPEN
B3:0
5
Ion pump 1 valve solenoid
IPV1SOL
O:0
L
2
1761-Micro
0017
IonPump2 valve open
IPV2_OPEN
B3:0
6
IonPump2 Valve Solenoid
IPV2SOL
O:0
L
3
1761-Micro
0018
Turbo up to speed intlk
UP_TO_SPEED
I:0
17
1761-Micro
turbo slowing down from speed
TURBO_SLOWING_DOWN
B3:0
9
0019
turbo slowing down from speed
TURBO_SLOWING_DOWN
B3:0
9
latches turbo up to speed state
UP_TO_SPEED_LATCH
B3:0
L
8

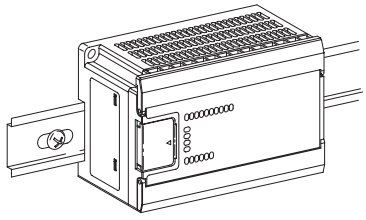

0020
latches turbo up to speed state
UP_TO_SPEED_LATCH
B3:0
8
TV Unlatch Delay
UNLATCH_DELAY
TON
Timer On Delay
Timer T4:1
Time Base 0.01
Preset 500<
Accum 500<
EN
DN
0021
TV Unlatch Delay
UNLATCH_DELAY/DN
T4:1
DN
latches turbo up to speed state
UP_TO_SPEED_LATCH
B3:0
U
8
0022
Turbo up to speed intlk
UP_TO_SPEED
I:0
17
1761-Micro
Major fault power failure
PWRFAIL
B3:0
11
Nitrogen solenoid
N2SOL
O:0
4
1761-Micro
0023
RESET for all latched valves
RESET
I:0
19
1761-Micro
reset buttuon delay
RESET_BUTTON
TON
Timer On Delay
Timer T4:0
Time Base 0.01
Preset 500<
Accum 0<
EN
DN


reset buttuon delay
RESET_BUTTON/DN
T4:0
DN
0024
Cryopump Valve Solenoid
CVSOL
O:0
U
1
1761-Micro
Turbo valve solenoid
TVSOL
O:0
U
0
1761-Micro
Ion pump 1 valve solenoid
IPV1SOL
O:0
U
2
1761-Micro
IonPump2 Valve Solenoid
IPV2SOL
O:0
U
3
1761-Micro
0025
END

Allen-Bradley

MicroLogix™ 1000 Programmable Controllers

(Bulletin 1761 Controllers)



User Manual

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes, and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based on the examples shown in this publication.

Allen-Bradley publication SGI-1.1, Safety Guidelines for the Application, Installation, and Maintenance of Solid-State Control (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted publication, in whole or in part, without written permission of Allen-Bradley Company, Inc., is prohibited.

Throughout this manual, we use notes to make you aware of safety considerations:

Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss.

Attention statements help you to:

- identify a hazard
- avoid the hazard
- recognize the consequences

Note *Identifies information that is critical for successful application and understanding of the product.*

SLC 500, SLC 5/01, SLC 5/02, SLC 5/03, SLC 5/04, MicroLogix, DTAM, DTAM Micro, PanelView, RediPANEL, Dataliner, DH+, and Data Highway Plus are trademarks of Rockwell Automation.
PLC-2, PLC-5 are registered trademarks of Rockwell Automation.
A.I. Series and WINtelligent LINX are trademarks of Rockwell Software Inc.

Table of Contents

Preface **P–1**
Who Should Use this Manual P–2
Purpose of this Manual P–2
Common Techniques Used in this Manual P–6
Allen-Bradley Support P–6

Hardware

1 Installing Your Controller **1–1**
Compliance to European Union Directives 1–2
Hardware Overview 1–3
Master Control Relay 1–4
Using Surge Suppressors 1–8
Safety Considerations 1–11
Power Considerations 1–12
Preventing Excessive Heat 1–13
Controller Spacing 1–14
Mounting the Controller 1–14

2 Wiring Your Controller **2–1**
Grounding Guidelines 2–2
Sinking and Sourcing Circuits 2–3
Wiring Recommendations 2–4
Wiring Diagrams, Discrete Input and Output Voltage Ranges 2–7
Analog Cable Recommendation 2–21
Minimizing Electrical Noise on Analog Controllers 2–21
Wiring Your Analog Channels 2–22
Analog Voltage and Current Input and Output Ranges 2–23
Wiring Your Controller for High–Speed Counter Applications 2–24

3 Connecting the System **3–1**
Connecting the DF1 Protocol 3–2
Connecting to a DH-485 Network 3–5
Connecting the AIC+ 3–9

Establishing Communication 3–17
DeviceNet Communications 3–18

Programming

4 Programming Overview 4–1
Principles of Machine Control 4–2
Understanding File Organization 4–4
Understanding How Processor Files are Stored and Accessed 4–6
Addressing Data Files 4–10
Applying Ladder Logics to Your Schematics 4–14
Developing Your Logic Program – A Model 4–15

5 Using Analog 5–1
I/O Image 5–2
I/O Configuration 5–3
Input Filter and Update Times 5–3
Converting Analog Data 5–5

6 Using Basic Instructions 6–1
About the Basic Instructions 6–2
Bit Instructions Overview 6–3
Examine if Closed (XIC) 6–4
Examine if Open (XIO) 6–4
Output Energize (OTE) 6–5
Output Latch (OTL) and Output Unlatch (OTU) 6–5
One-Shot Rising (OSR) 6–7
Timer Instructions Overview 6–8
Timer On-Delay (TON) 6–11
Timer Off-Delay (TOF) 6–12
Retentive Timer (RTO) 6–14
Counter Instructions Overview 6–15
Count Up (CTU) 6–18
Count Down (CTD) 6–19
Reset (RES) 6–20
Basic Instructions in the Paper Drilling Machine Application Example 6–21

7 Using Comparison Instructions 7–1
About the Comparison Instructions 7–2
Comparison Instructions Overview 7–2

Equal (EQU) 7–3
Not Equal (NEQ) 7–3
Less Than (LES) 7–3
Less Than or Equal (LEQ) 7–4
Greater Than (GRT) 7–4
Greater Than or Equal (GEQ) 7–4
Masked Comparison for Equal (MEQ) 7–5
Limit Test (LIM) 7–6
Comparison Instructions in the Paper Drilling Machine Application Example 7–8

8 Using Math Instructions 8–1
About the Math Instructions 8–2
Math Instructions Overview 8–2
Add (ADD) 8–4
Subtract (SUB) 8–5
32-Bit Addition and Subtraction 8–6
Multiply (MUL) 8–8
Divide (DIV) 8–9
Double Divide (DDV) 8–10
Clear (CLR) 8–11
Square Root (SQR) 8–11
Scale Data (SCL) 8–12
Math Instructions in the Paper Drilling Machine Application Example 8–14

9 Using Data Handling Instructions 9–1
About the Data Handling Instructions 9–2
Convert to BCD (TOD) 9–3
Convert from BCD (FRD) 9–5
Decode 4 to 1 of 16 (DCD) 9–8
Encode 1 of 16 to 4 (ENC) 9–9
Copy File (COP) and Fill File (FLL) Instructions 9–10
Move and Logical Instructions Overview 9–13
Move (MOV) 9–15
Masked Move (MVM) 9–16
And (AND) 9–18
Or (OR) 9–19
Exclusive Or (XOR) 9–20
Not (NOT) 9–21
Negate (NEG) 9–22
FIFO and LIFO Instructions Overview 9–23
FIFO Load (FFL) and FIFO Unload (FFU) 9–25
LIFO Load (LFL) and LIFO Unload (LFU) 9–26

Data Handling Instructions in the Paper Drilling Machine Application Example 9–28

10 Using Program Flow Control Instructions 10–1

About the Program Flow Control Instructions 10–2
Jump (JMP) and Label (LBL) 10–2
Jump to Subroutine (JSR), Subroutine (SBR), and Return (RET) 10–4
Master Control Reset (MCR) 10–7
Temporary End (TND) 10–8
Suspend (SUS) 10–8
Immediate Input with Mask (IIM) 10–9
Immediate Output with Mask (IOM) 10–9
Program Flow Control Instructions in the Paper Drilling Machine Application Example 10–10

11 Using Application Specific Instructions 11–1

About the Application Specific Instructions 11–2
Bit Shift Instructions Overview 11–3
Bit Shift Left (BSL) 11–5
Bit Shift Right (BSR) 11–6
Sequencer Instructions Overview 11–7
Sequencer Output (SQO) and Sequencer Compare (SQC) 11–7
Sequencer Load (SQL) 11-13
Selectable Timed Interrupt (STI) Function Overview 11–15
Selectable Timed Disable (STD) and Enable (STE) 11–18
Selectable Timed Start (STS) 11–20
Interrupt Subroutine (INT) 11–20
Application Specific Instructions in the Paper Drilling Machine Application Example 11–21

12 Using High-Speed Counter Instructions 12–1

About the High-Speed Counter Instructions 12–2
High-Speed Counter Instructions Overview 12–3
High-Speed Counter (HSC) 12–6
High-Speed Counter Load (HSL) 12-18
High-Speed Counter Reset (RES) 12–21
High-Speed Counter Reset Accumulator (RAC) 12–22
High-Speed Counter Interrupt Enable (HSE) and Disable (HSD) 12–23
Update High-Speed Counter Image Accumulator (OTE) 12–24
What Happens to the HSC When Going to REM Run Mode 12–25
High-Speed Counter Instructions in the Paper Drilling Machine Application Example 12–29

13 Using the Message Instruction 13–1
Types of Communication 13–2
Message Instruction (MSG) 13–3
Timing Diagram for a Successful MSG Instruction 13–8
MSG Instruction Error Codes 13–10
Application Examples that Use the MSG Instruction 13–12

Troubleshooting

14 Troubleshooting Your System 14–1
Understanding the Controller LED Status 14–2
Controller Error Recovery Model 14–5
Identifying Controller Faults 14–6
Calling Allen-Bradley for Assistance 14–10

Reference

A Hardware Reference A–1
Controller Specifications A–2
Controller Dimensions A–9
Replacement Parts A–10

B Programming Reference B–1
Controller Status File B–1
Instruction Execution Times and Memory Usage B–21

C Valid Addressing Modes and File Types for Instruction Parameters C–1
Available File Types C–2
Available Addressing Modes C–3

D Understanding the Communication Protocols D–1
RS-232 Communication Interface D–2
DF1 Full-Duplex Protocol D–3
DF1 Half-Duplex Slave Protocol D–5
DH-485 Communication Protocol D–11

E Application Example Programs E–1
Paper Drilling Machine Application Example E–2
Time Driven Sequencer Application Example E–17
Event Driven Sequencer Application Example E–19
Bottle Line Example E–21
Pick and Place Machine Example E–24
RPM Calculation Application Example E–28
On/Off Circuit Application Example E–34
Spray Booth Application Example E–36
Adjustable Timer Application Example E–41

F Optional Analog Input Software Calibration F–1
Calibrating an Analog Input Channel F–2

Glossary G–1

Summary of Changes

The information below summarizes the changes to this manual since the last printing as Publication 1761-6.3 — December 1997.

To help you find new information and updated information in this release of the manual, we have included change bars as shown to the right of this paragraph.

New Information

The table below lists sections that document new features and additional information about existing features, and shows where to find this new information.

For This New Information	See
Power supply inrush	page 1–13

Updated Information

Changes from the previous release of this manual that require you to reference information differently are as follows:

- The DeviceNet communications information has been updated; see chapter 3, Connecting the System.
- For updated information on automatic protocol switching, see chapter 3, Connecting the System.
- The MicroLogix 1000 programmable controllers' VA ratings and power supply inrush specifications have been updated; see appendix A, Hardware Reference.
- The DF1 Full-Duplex and DH-485 configuration parameters have been updated; see appendix D, Understanding Communication Protocols.

Notes:

Preface

Read this preface to familiarize yourself with the rest of the manual. It provides information concerning:

- who should use this manual
- the purpose of this manual
- conventions used in this manual
- Allen-Bradley support

Who Should Use this Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use MicroLogix™ 1000 controllers.

You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

Purpose of this Manual

This manual is a reference guide for MicroLogix 1000 controllers. It describes the procedures you use to install, wire, program, and troubleshoot your controller. This manual:

- explains how to install and wire your controllers
- gives you an overview of the MicroLogix 1000 controller system
- provides the MicroLogix 1000 controllers' instruction set
- contains application examples to show the instruction set in use

See your programming software user manual for information on programming your MicroLogix 1000 controller. For information on using the Hand-Held Programmer with the MicroLogix 1000 controllers, see the *MicroLogix 1000 with Hand-Held Programmer (HHP) User Manual*, Publication 1761-6.2.

Contents of this Manual

Tab	Chapter	Title	Contents
		Preface	Describes the purpose, background, and scope of this manual. Also specifies the audience for whom this manual is intended.
Hardware	1	Installing Your Controller	Provides controller installation procedures and system safety considerations.
	2	Wiring Your Controller	Provides wiring guidelines and diagrams.
	3	Connecting the System	Gives information on wiring your controller system for the DF1 protocol or DH-485 network.
Programming	4	Programming Overview	Provides an overview of principles of machine control, a section on file organization and addressing, and a program development model.
	5	Using Analog	Provides information on I/O image file format, I/O configuration, input filter and update times, and conversion of analog data.
	6	Using Basic Instructions	Describes how to use ladder logic instructions for relay replacement functions, counting, and timing.
	7	Using Comparison Instructions	Describes how to use the instructions to compare values of data in your ladder logic program.
	8	Using Math Instructions	Describes how to use the ladder logic instructions that perform basic math functions.
	9	Using Data Handling Instructions	Describes how to perform data handling instructions, including move and logical instructions and FIFO and LIFO instructions.
	10	Using Program Flow Control Instructions	Describes the ladder logic instructions that affect program flow and execution.
	11	Using Application Specific Instructions	Describes the bit shift, sequencer and STI related instructions.
	12	Using High-Speed Counter Instructions	Describes the four modes of the high-speed counter and its related instructions.
	13	Using the Message Instruction	Provides a general overview of the types of communication, and explains how to establish network communication using the message instruction.
Troubleshooting	14	Troubleshooting Your System	Explains how to interpret and correct problems with your MicroLogix 1000 controller system.

Tab	Chapter	Title	Contents
Reference	Appendix A	Hardware Reference	Provides physical, electrical, environmental, and functional specifications.
	Appendix B	Programming Reference	Explains the system status file and provides instruction execution times.
	Appendix C	Valid Addressing Modes and File Types for Instruction Parameters	Provides a listing of the instructions along with their parameters and valid file types.
	Appendix D	Understanding the Communication Protocols	Contains descriptions of the DF1 protocol and DH-485 network.
	Appendix E	Application Example Programs	Provides advanced application examples for the high-speed counter, sequencer, bit shift, and message instructions.
	Appendix F	Optional Analog Input Software Calibration	Explains how to calibrate your controller using software offsets.
		Glossary	Contains definitions for terms and abbreviations that are specific to this product.

Related Documentation

The following documents contain additional information concerning Allen-Bradley products. To obtain a copy, contact your local Allen-Bradley office or distributor.

For	Read this Document	Document Number
A procedural manual for technical personnel who use the Allen-Bradley Hand-Held Programmer (HHP) to monitor and develop control logic programs for the MicroLogix 1000 controller.	MicroLogix™ 1000 with Hand-Held Programmer (HHP) User Manual	1761-6.2
Information on mounting and wiring the MicroLogix 1000 controllers, including a mounting template for easy installation	MicroLogix™ 1000 Programmable Controllers Installation Instructions	1761-5.1.2
	MicroLogix™ 1000 (Analog) Programmable Controllers Installation Instructions	1761-5.1.3
The procedures necessary to install and connect the AIC+ and DNI	Advanced Interface Converter (AIC+) and DeviceNet Interface (DNI) Installation Instructions	1761-5.11
A description on how to install and connect an AIC+. This manual also contains information on network wiring.	Advanced Interface Converter (AIC+) User Manual	1761-6.4
Information on how to install, configure, and commission a DNI	DeviceNet Interface™ User Manual	1761-6.5
In-depth information on grounding and wiring Allen-Bradley programmable controllers	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1
A description of important differences between solid-state programmable controller products and hard-wired electromechanical devices	Application Considerations for Solid-State Controls	SGI-1.1
An article on wire sizes and types for grounding electrical equipment	National Electrical Code	Published by the National Fire Protection Association of Boston, MA.
A complete listing of current documentation, including ordering instructions. Also indicates whether the documents are available on CD-ROM or in multi-languages.	Allen-Bradley Publication Index	SD499
A glossary of industrial automation terms and abbreviations	Allen-Bradley Industrial Automation Glossary	AG-7.1
Information on understanding and applying MicroLogix 1000 controllers	MicroMentor	1761-MMB

Common Techniques Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.

Allen-Bradley Support

Allen-Bradley offers support services worldwide, with over 75 Sales/Support Offices, 512 authorized Distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Allen-Bradley representatives in every major country in the world.

Local Product Support

Contact your local Allen-Bradley representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Allen-Bradley for technical assistance, please review the information in the *Troubleshooting* chapter first. Then call your local Allen-Bradley representative.

Your Questions or Comments on this Manual

If you find a problem with this manual, or you have any suggestions for how this manual could be made more useful to you, please contact us at the address below:

Allen-Bradley Company, Inc.
Control and Information Group
Technical Communication, Dept. 602V, T122
P.O. Box 2086
Milwaukee, WI 53201-2086

or visit our internet page at:

http://www.ab.com/micrologix

1 Installing Your Controller

This chapter shows you how to install your controller system. The only tools you require are a Flat head or Phillips head screwdriver and drill. Topics include:

- compliance to European Union Directives
- hardware overview
- master control relay
- surge suppressors
- safety considerations
- power considerations
- preventing excessive heat
- controller spacing
- mounting the controller

Compliance to European Union Directives

If this product has the CE mark it is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

This product is tested to meet Council Directive 89/336/EEC Electromagnetic Compatibility (EMC) and the following standards, in whole or in part, documented in a technical construction file:

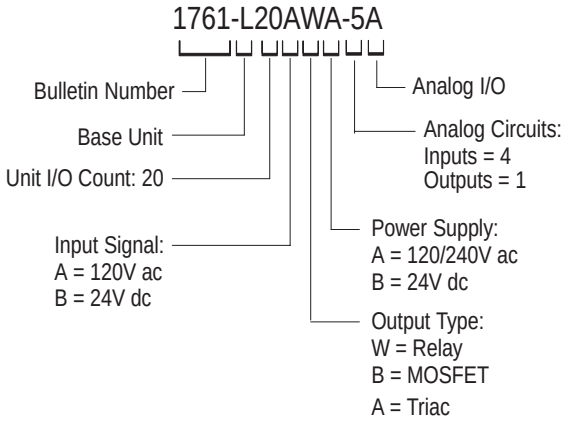
- EN 50081-2
  EMC – Generic Emission Standard, Part 2 – Industrial Environment
- EN 50082-2
  EMC – Generic Immunity Standard, Part 2 – Industrial Environment

This product is intended for use in an industrial environment.

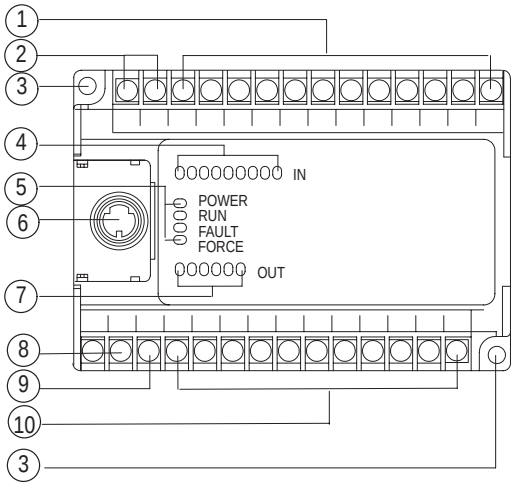
Hardware Overview

The MicroLogix 1000 programmable controller is a packaged controller containing a power supply, input circuits, output circuits, and a processor. The controller is available in 10 I/O, 16 I/O and 32 I/O configurations, as well as an analog version with 20 discrete I/O and 5 analog I/O.

The catalog number for the controller is composed of the following:



The hardware features of the controller are:



1. Input terminals
2. dc output terminals (or not used)
3. Mounting hole
4. Input LEDs
5. Status LEDs
6. RS-232 communication channel
7. Output LEDs
8. Power supply line power
9. Ground screw
10. Output terminals

20142

Master Control Relay

A hard-wired master control relay (MCR) provides a reliable means for emergency controller shutdown. Since the master control relay allows the placement of several emergency-stop switches in different locations, its installation is important from a safety standpoint. Overtravel limit switches or mushroom head push buttons are wired in series so that when any of them opens, the master control relay is de-energized. This removes power to input and output device circuits. Refer to the figure on page 1–6.

Never alter these circuits to defeat their function, since serious injury and/or machine damage could result.

Note *If you are using an external dc output power supply, interrupt the dc output side rather than the ac line side of the supply to avoid the additional delay of power supply turn-off.*

The external ac line of the dc output power supply should be fused.

Connect a set of master control relays in series with the dc power supplying the input and output circuits.

Place the main power disconnect switch where operators and maintenance personnel have quick and easy access to it. If you mount a disconnect switch inside the controller enclosure, place the switch operating handle on the outside of the enclosure, so that you can disconnect power without opening the enclosure.

Whenever any of the emergency-stop switches are opened, power to input and output devices should be removed.

When you use the master control relay to remove power from the external I/O circuits, power continues to be provided to the controller's power supply so that diagnostic indicators on the processor can still be observed.

The master control relay is not a substitute for a disconnect to the controller. It is intended for any situation where the operator must quickly de-energize I/O devices only. When inspecting or installing terminal connections, replacing output fuses, or working on equipment within the enclosure, use the disconnect to shut off power to the rest of the system.

Note *Do not control the master control relay with the controller. Provide the operator with the safety of a direct connection between an emergency-stop switch and the master control relay.*

Using Emergency-Stop Switches

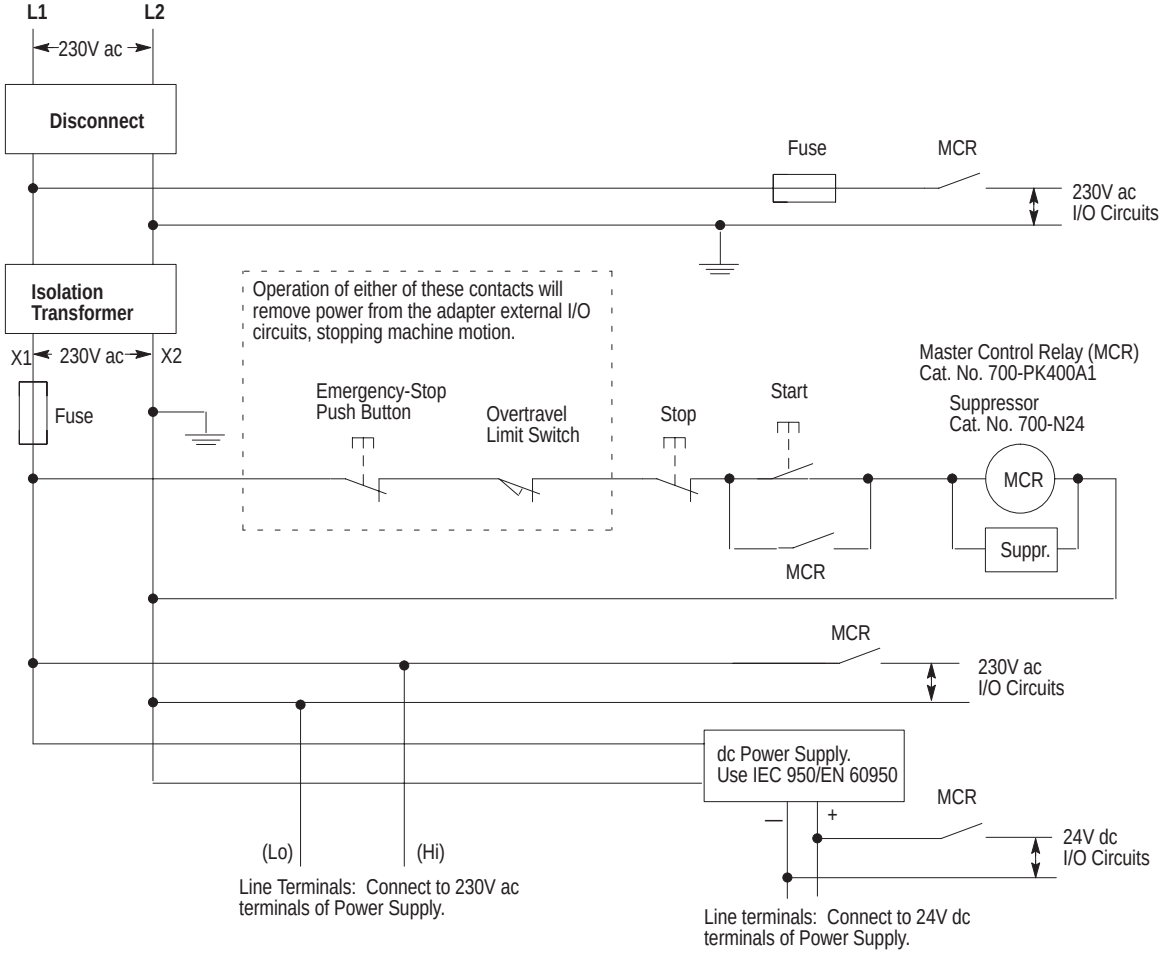
When using emergency-stop switches, adhere to the following points:

- Do not program emergency-stop switches in the controller program. Any emergency-stop switch should turn off all machine power by turning off the master control relay.
- Observe all applicable local codes concerning the placement and labeling of emergency-stop switches.
- Install emergency-stop switches and the master control relay in your system. Make certain that relay contacts have a sufficient rating for your application. Emergency-stop switches must be easy to reach.
- In the following illustration, input and output circuits are shown with MCR protection. However, in most applications, only output circuits require MCR protection.

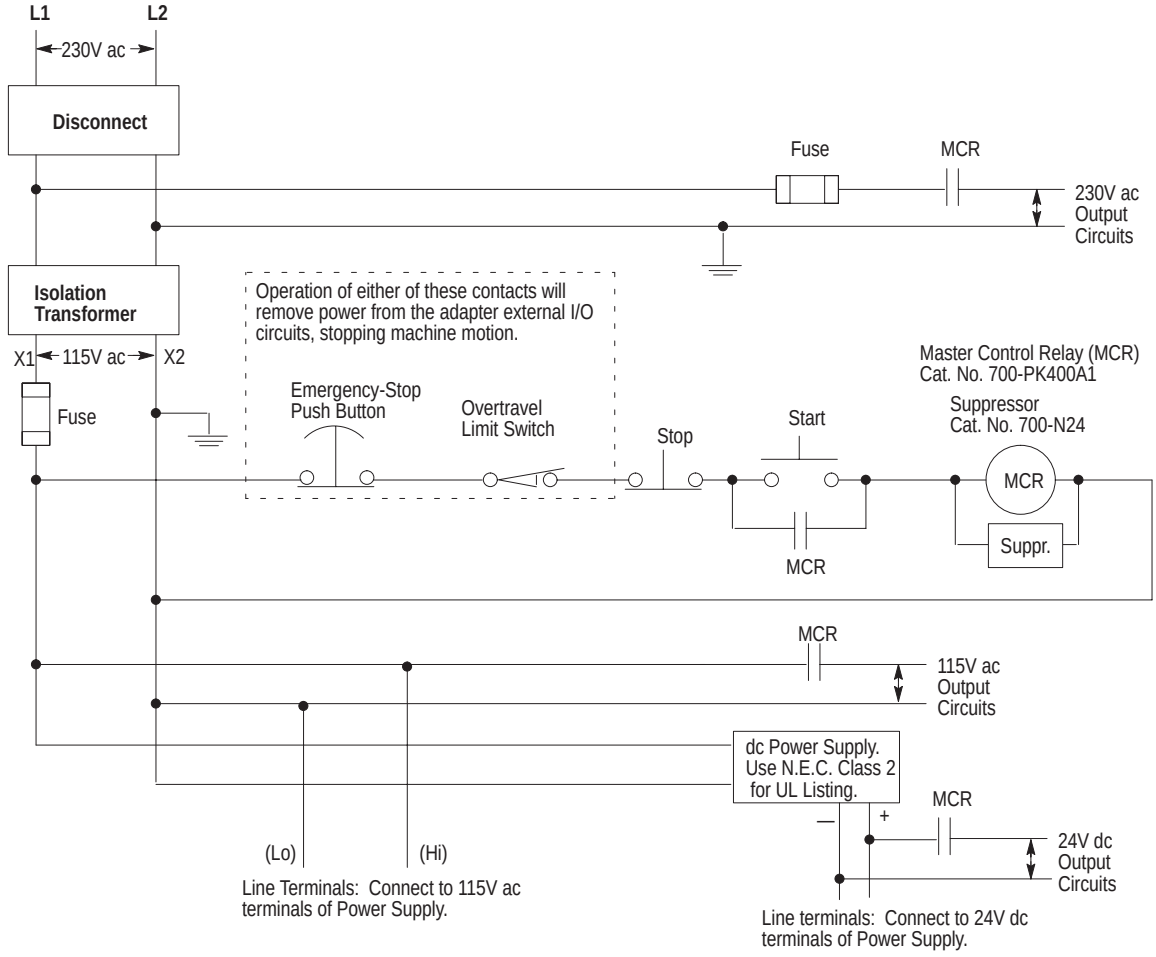
The following illustrations show the Master Control Relay wired in a grounded system.

Note *The illustrations only show output circuits with MCR protection. In most applications input circuits do not require MCR protection; however, if you need to remove power from all field devices, you must include MCR contacts in series with input power wiring.*

Schematic (Using IEC Symbols)



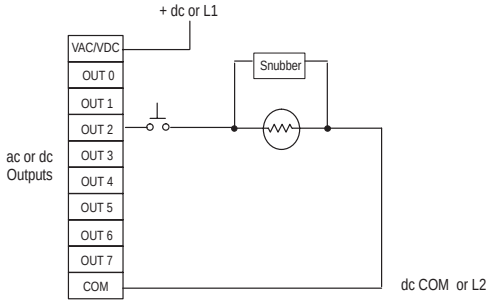
Schematic (Using ANSI/CSA Symbols)



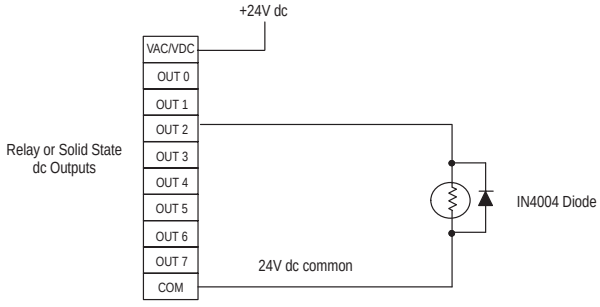
Using Surge Suppressors

Inductive load devices such as motor starters and solenoids require the use of some type of surge suppression to protect the controller output contacts. Switching inductive loads without surge suppression can *significantly* reduce the lifetime of relay contacts. By adding a suppression device directly across the coil of an inductive device, you will prolong the life of the switch contacts. You will also reduce the effects of voltage transients caused by interrupting the current to that inductive device, and will prevent electrical noise from radiating into system wiring.

The following diagram shows an output with a suppression device. We recommend that you locate the suppression device as close as possible to the load device.

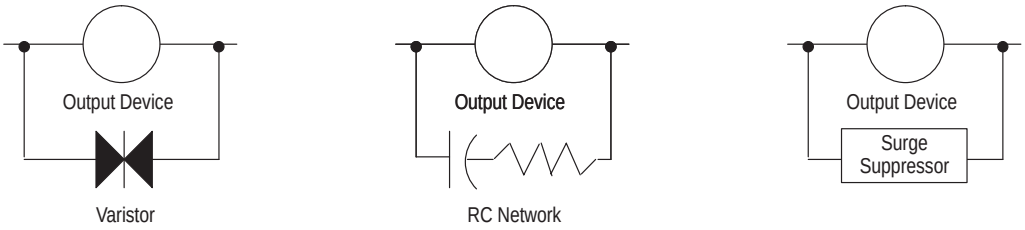


If you connect a micro controller FET output to an inductive load, we recommend that you use an 1N4004 diode for surge suppression, as shown in the illustration that follows.



Suitable surge suppression methods for inductive ac load devices include a varistor, an RC network, or an Allen-Bradley surge suppressor, all shown below. These components must be appropriately rated to suppress the switching transient characteristic of the particular inductive device. See the table on page 1–10 for recommended suppressors.

Surge Suppression for Inductive ac Load Devices

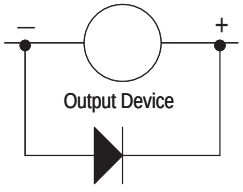


If you connect a micro controller triac output to control an inductive load, we recommend that you use varistors to suppress noise. Choose a varistor that is appropriate for the application. The suppressors we recommend for triac outputs when switching 120V ac inductive loads are a Harris MOV, part number V175 LA10A, or an Allen-Bradley MOV, catalog number 599-K04 or 599-KA04. Consult the varistor manufacturer's data sheet when selecting a varistor for your application.

For inductive dc load devices, a diode is suitable. An 1N4004 diode is acceptable for most applications. A surge suppressor can also be used. See the table on page 1–10 for recommended suppressors.

As shown in the illustration below, these surge suppression circuits connect directly across the load device. This reduces arcing of the output contacts. (High transient can cause arcing that occurs when switching off an inductive device.)

Surge Suppression for Inductive dc Load Devices



Diode
(A surge suppressor can also be used.)

Recommended Surge Suppressors

We recommend the Allen-Bradley surge suppressors shown in the following table for use with Allen-Bradley relays, contactors, and starters.

Device	Coil Voltage	Suppressor Catalog Number
Bulletin 509 Motor Starter Bulletin 509 Motor Starter	120V ac 240V ac	599-K04 599-KA04
Bulletin 100 Contactor Bulletin 100 Contactor	120V ac 240V ac	199-FSMA1 199-FSMA2
Bulletin 709 Motor Starter	120V ac	1401-N10
Bulletin 700 Type R, RM Relays	ac coil	None Required
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	12V dc 12V dc	700-N22 700-N28
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	24V dc 24V dc	700-N10 700-N13
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	48V dc 48V dc	700-N16 700-N17
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	115-125V dc 115-125V dc	700-N11 700-N14
Bulletin 700 Type R Relay Bulletin 700 Type RM Relay	230-250V dc 230-250V dc	700-N12 700-N15
Bulletin 700 Type N, P, or PK Relay	150V max, ac or DC	700-N24
Miscellaneous electromagnetic devices limited to 35 sealed VA	150V max, ac or DC	700-N24

Safety Considerations

Safety considerations are an important element of proper system installation. Actively thinking about the safety of yourself and others, as well as the condition of your equipment, is of primary importance. We recommend reviewing the following safety considerations.

Disconnecting Main Power

Explosion Hazard — Do not replace components or disconnect equipment unless power has been switched off and the area is known to be non-hazardous.

The main power disconnect switch should be located where operators and maintenance personnel have quick and easy access to it. In addition to disconnecting electrical power, all other sources of power (pneumatic and hydraulic) should be de-energized before working on a machine or process controlled by a controller.

Safety Circuits

Explosion Hazard — Do not connect or disconnect connectors while circuit is live unless area is known to be non-hazardous.

Circuits installed on the machine for safety reasons, like overtravel limit switches, stop push buttons, and interlocks, should always be hard-wired directly to the master control relay. These devices must be wired in series so that when any one device opens, the master control relay is de-energized thereby removing power to the machine. Never alter these circuits to defeat their function. Serious injury or machine damage could result.

Power Distribution

There are some points about power distribution that you should know:

- The master control relay must be able to inhibit all machine motion by removing power to the machine I/O devices when the relay is de-energized.
- If you are using a dc power supply, interrupt the load side rather than the ac line power. This avoids the additional delay of power supply turn-off. The dc power supply should be powered directly from the fused secondary of the transformer. Power to the dc input and output circuits is connected through a set of master control relay contacts.

Periodic Tests of Master Control Relay Circuit

Any part can fail, including the switches in a master control relay circuit. The failure of one of these switches would most likely cause an open circuit, which would be a safe power-off failure. However, if one of these switches shorts out, it no longer provides any safety protection. These switches should be tested periodically to assure they will stop machine motion when needed.

Power Considerations

The following explains power considerations for the micro controllers.

Isolation Transformers

You may want to use an isolation transformer in the ac line to the controller. This type of transformer provides isolation from your power distribution system and is often used as a step down transformer to reduce line voltage. Any transformer used with the controller must have a sufficient power rating for its load. The power rating is expressed in volt-amperes (VA).

Power Supply Inrush

The MicroLogix power supply does not require or need a high inrush current. However, if the power source can supply a high inrush current, the MicroLogix power supply will accept it. There is a high level of inrush current when a large capacitor on the input of the MicroLogix is charged up quickly.

If the power source cannot supply high inrush current, the only effect is that the MicroLogix input capacitor charges up more slowly. The following considerations determine whether the power source needs to supply a high inrush current:

- power-up sequence of devices in system
- power source sag if it cannot source inrush current
- the effect of the voltage sag on other equipment

If the power source cannot provide high inrush current when the entire system in an application is powered, the MicroLogix powers-up more slowly. If part of an application's system is already powered and operating when the MicroLogix is powered, the source voltage may sag while the MicroLogix input capacitor is charging. A power source voltage sag can affect other equipment connected to the same power source. For example, a voltage sag may reset a computer connected to the same power source.

Loss of Power Source

The power supply is designed to withstand brief power losses without affecting the operation of the system. The time the system is operational during power loss is called "program scan hold-up time after loss of power." The duration of the power supply hold-up time depends on the type and state of the I/O, but is typically between 20 milliseconds and 3 seconds. When the duration of power loss reaches this limit, the power supply signals the processor that it can no longer provide adequate dc power to the system. This is referred to as a power supply shutdown.

Input States on Power Down

The power supply hold-up time as described above is generally longer than the turn-on and turn-off times of the inputs. Because of this, the input state change from "On" to "Off" that occurs when power is removed may be recorded by the processor before the power supply shuts down the system. Understanding this concept is important. The user program should be written to take this effect into account.

Other Types of Line Conditions

Occasionally the power source to the system can be temporarily interrupted. It is also possible that the voltage level may drop substantially below the normal line voltage range for a period of time. Both of these conditions are considered to be a loss of power for the system.

Preventing Excessive Heat

For most applications, normal convective cooling keeps the controller within the specified operating range. Ensure that the specified operating range is maintained. Proper spacing of components within an enclosure is usually sufficient for heat dissipation.

In some applications, a substantial amount of heat is produced by other equipment inside or outside the enclosure. In this case, place blower fans inside the enclosure to assist in air circulation and to reduce "hot spots" near the controller.

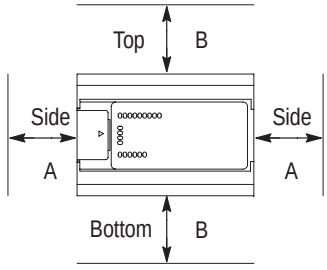
Additional cooling provisions might be necessary when high ambient temperatures are encountered.

Note *Do not bring in unfiltered outside air. Place the controller in an enclosure to protect it from a corrosive atmosphere. Harmful contaminants or dirt could cause improper operation or damage to components. In extreme cases, you may need to use air conditioning to protect against heat build-up within the enclosure.*

Controller Spacing

The following figure shows the recommended *minimum* spacing for the controller. (Refer to appendix A for controller dimensions.)

Explosion Hazard — For Class I, Division 2 applications, this product must be installed in an enclosure. All cables connected to the product must remain in the enclosure or be protected by conduit or other means.



A. Greater than or equal to 50.8 mm (2 in.).

B. Greater than or equal to 50.8 mm (2 in.).

20142

Mounting the Controller

This equipment is suitable for Class I, Division 2, Groups A, B, C, D or non-hazardous locations only, when product or packaging is marked.

Explosion Hazard:

- **Substitution of components may impair suitability for Class I, Division 2.**
- **Be careful of metal chips when drilling mounting holes for your controller. Drilled fragments that fall into the controller could cause damage. Do not drill holes above a mounted controller if the protective wrap is removed.**

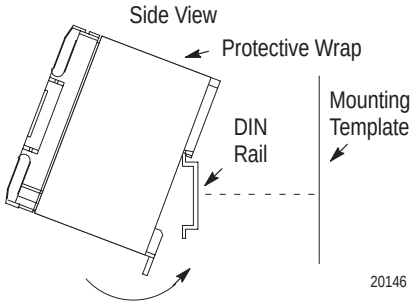
The controller should be mounted horizontally within an enclosure, using a DIN rail or mounting screws.

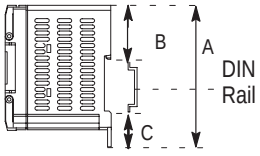
Using a DIN Rail

Use 35 mm (1.38 in.) DIN rails, such as item number 199-DR1 or 1492-DR5 from Bulletin 1492.

To install your controller on the DIN rail:

1. Mount your DIN rail. (Make sure that the placement of the controller on the DIN rail meets the recommended spacing requirements. Refer to controller dimensions in appendix A.)
2. Hook the top slot over the DIN rail.
3. While pressing the controller against the rail, snap the controller into position.
4. Leave the protective wrap attached until you are finished wiring the controller.

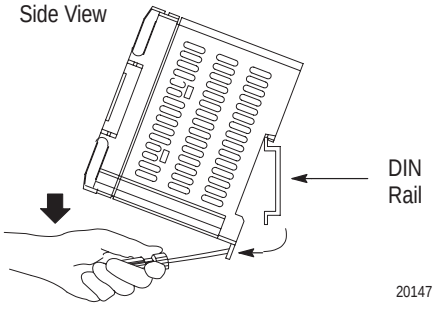




Call-out	Dimension
A	84 mm (3.3 in.)
B	33 mm (1.3 in.)
C	16 mm (.63 in.)

To remove your controller from the DIN rail:

1. Place a screwdriver in the DIN rail latch at the bottom of the controller.
2. Holding the controller, pry downward on the latch until the controller is released from the DIN rail.

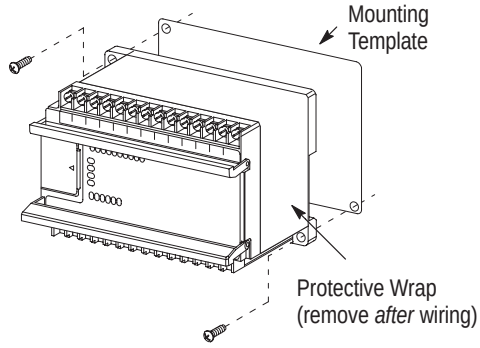


Using Mounting Screws

To install your controller using mounting screws:

Note Leave the protective wrap attached until you are finished wiring the controller.

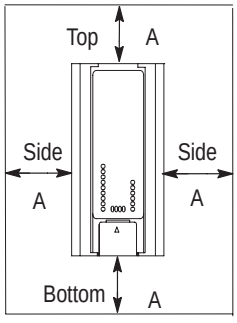
1. Use the mounting template from the *MicroLogix 1000 Programmable Controllers Installation Instructions*, publication 1761-5.1.2 or *MicroLogix 1000 (Analog) Programmable Controllers Installation Instructions*, publication 1761-5.1.3, that was shipped with your controller.
2. Secure the template to the mounting surface. (Make sure your controller is spaced properly.)
3. Drill holes through the template.
4. Remove the mounting template.
5. Mount the controller.



Mounting Your Controller Vertically

Your controller can also be mounted vertically within an enclosure using mounting screws or a DIN rail. To insure the stability of your controller, we recommend using mounting screws.

To insure the controller's reliability, the following environmental specifications must not be exceeded.



A. Greater than or equal to 50.8 mm (2 in.).

Description:	Specification:
Operating Temperature	Discrete: 0°C to +45°C (+32°F to +113°F)[1] Analog: 0°C to +40°C (+32°F to +113°F)[1]
Operating Shock (Panel mounted)	9.0g peak acceleration (11±1 ms duration) 3 times each direction, each axis
Operating Shock (DIN rail mounted)	7.0g peak acceleration (11±1 ms duration) 3 times each direction, each axis

[1] DC input voltage derated linearly from +30°C (30V to 26.4V).

Note: When mounting your controller vertically, the nameplate should be facing downward.

2 Wiring Your Controller

This chapter describes how to wire your controller. Topics include:

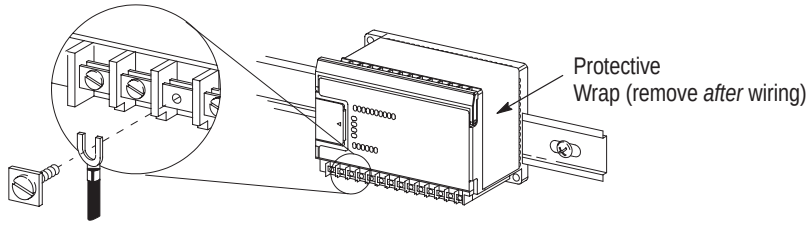
- grounding guidelines
- sinking and sourcing circuits
- wiring recommendations
- wiring diagrams, input voltage ranges, and output voltage ranges

Grounding Guidelines

In solid-state control systems, grounding helps limit the effects of noise due to electromagnetic interference (EMI). Use the heaviest wire gauge listed for wiring your controller with a maximum length of 152.4 mm (6 in.). Run the ground connection from the ground screw of the controller (third screw from left on output terminal rung) to the ground bus.

Note

⏚ *This symbol denotes a functional earth ground terminal which provides a low impedance path between electrical circuits and earth for non-safety purposes, such as noise immunity improvement.*



All devices that connect to the user 24V power supply or to the RS-232 channel must be referenced to chassis ground or floating. Failure to follow this procedure may result in property damage or personal injury.

Chassis ground, user 24V ground, and RS-232 ground are internally connected. You must connect the chassis ground terminal screw to chassis ground prior to connecting any devices.

On the 1761-L10BWB, 1761-L16BWB, 1761-L16BBB, 1761-L20BWB-5A, 1761-L32BBB, and 1761-L32BWB controllers, the user supply 24 V dc IN and chassis ground are internally connected.

You must also provide an acceptable grounding path for each device in your application. For more information on proper grounding guidelines, see the *Industrial Automation Wiring and Grounding Guidelines* publication 1770-4.1.

Remove the protective wrap before applying power to the controller. Failure to remove the wrap may cause the controller to overheat.

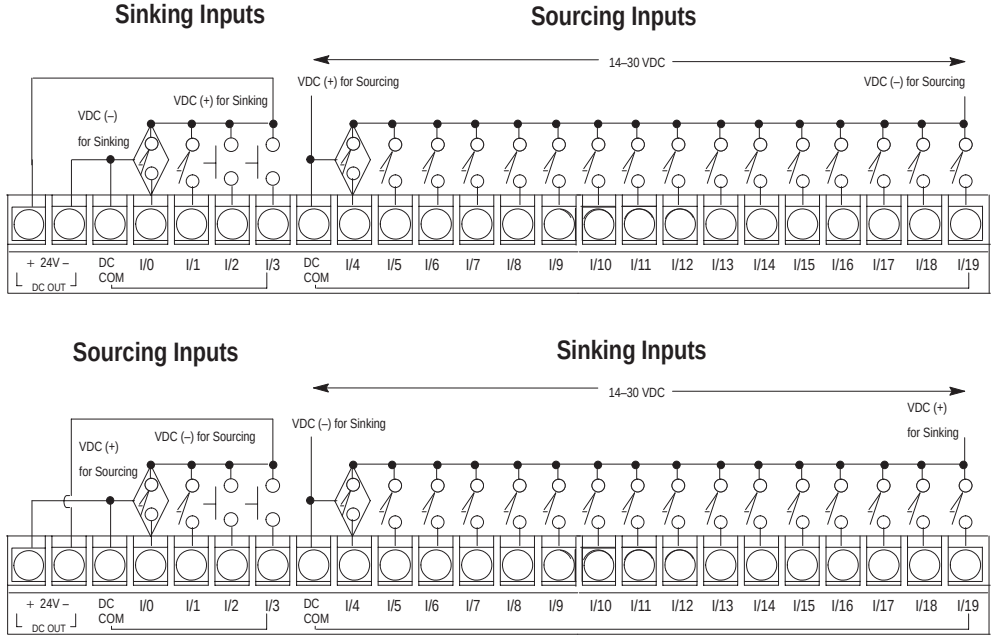
Sinking and Sourcing Circuits

Any of the MicroLogix 1000 DC inputs can be configured as sinking or sourcing depending on how the DC COM is wired on the MicroLogix.

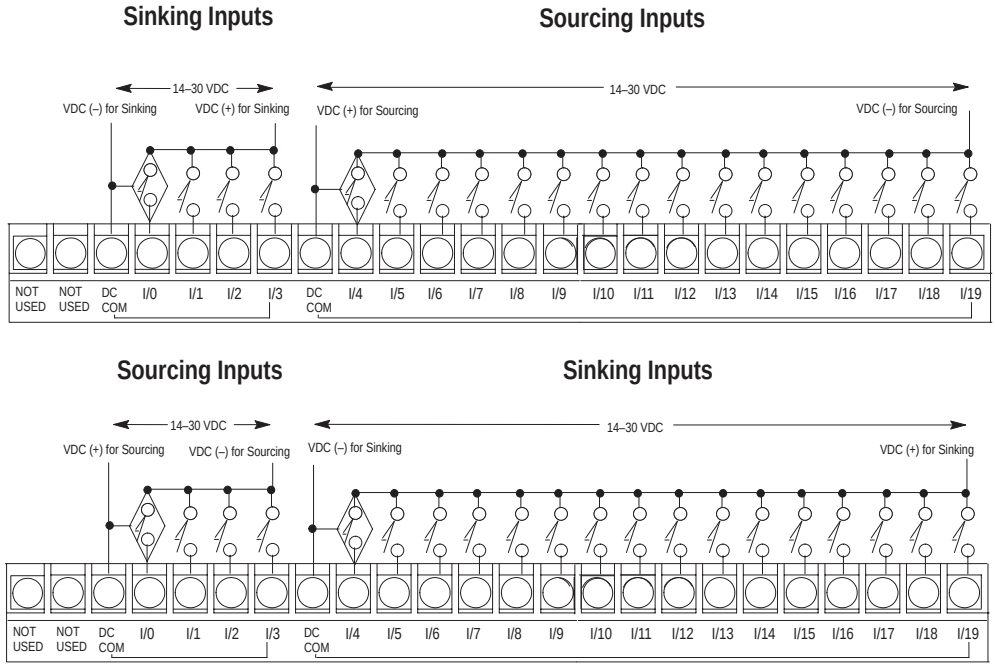
Type	Definition
Sinking Input	The input energizes when high-level voltage is applied to the input terminal (active high). Connect the power supply VDC (–) to the MicroLogix DC COM terminal.
Sourcing Input	The input energizes when low-level voltage is applied to the input terminal (active low). Connect the power supply VDC (+) to the MicroLogix DC COM terminal.

Sinking and Sourcing Wiring Examples

1761-L32BWA (Wiring diagrams also apply to 1761-L20BWA-5A, -L16BWA, -L10BWA.)



1761-L32BWB, -L32BBB (Wiring Diagrams also apply to 1761-L20BWB-5A, -L16BWB, -L10BWB, -L16BBB.)



Wiring Recommendations

Before you install and wire any device, disconnect power to the controller system.

The following are general recommendations for wiring your controller system.

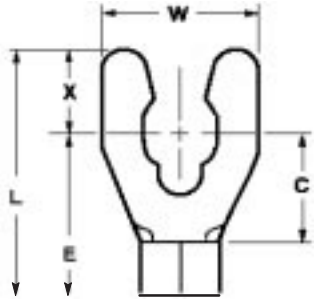
- Each wire terminal accepts 2 wires of the size listed below:

Wire Type	Wire Size (2 wire maximum per terminal screw)
Solid	#14 to #22 AWG
Stranded	#16 to #22 AWG

Refer to page 2–24 for wiring your high-speed counter.

Note

The diameter of the terminal screw heads is 5.5 mm (0.220 in.). The input and output terminals of the micro controller are designed for the following spade lugs:

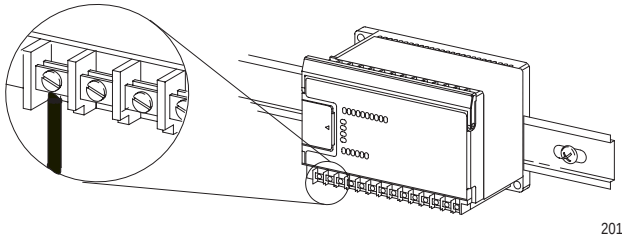


Call-out	Dimension
C	6.35 mm (0.250 in.)
E	10.95 mm (0.431 in.) maximum
L	14.63 mm (0.576 in.) maximum
W	6.35 mm (0.250 in.)
X	3.56 mm (0.140 in.)
C+X	9.91 mm (0.390 in.) maximum

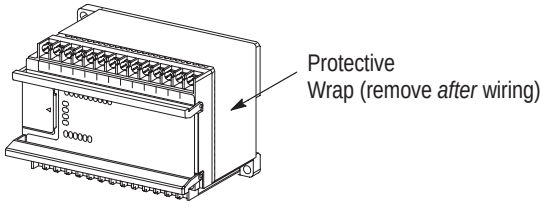
We recommend using either of the following AMP spade lugs: part number 53120-1, if using 22–16 AWG, or part number 53123-1, if using 16–14 AWG.

Note

If you use wires without lugs, make sure the wires are securely captured by the pressure plate. This is particularly important at the four end terminal positions where the pressure plate does not touch the outside wall.



Be careful when stripping wires. Wire fragments that fall into the controller could cause damage. Do not strip wires above a mounted controller if the protective wrap is removed.



Remove the protective wrap before applying power to the controller. Failure to remove the wrap may cause the controller to overheat.

Calculate the maximum possible current in each power and common wire. Observe all electrical codes dictating the maximum current allowable for each wire size. Current above the maximum ratings may cause wiring to overheat, which can cause damage.

United States Only: **If the controller is installed within a potentially hazardous environment, all wiring must comply with the requirements stated in the National Electrical Code 501-4 (b).**

- Allow for at least 50 mm (2 in.) between I/O wiring ducts or terminal strips and the controller.
- Route incoming power to the controller by a path separate from the device wiring. Where paths must cross, their intersection should be perpendicular.

Note

Do not run signal or communications wiring and power wiring in the same conduit. Wires with different signal characteristics should be routed by separate paths.

- Separate wiring by signal type. Bundle wiring with similar electrical characteristics together.
- Separate input wiring from output wiring.
- Label wiring to all devices in the system. Use tape, shrink-tubing, or other dependable means for labeling purposes. In addition to labeling, use colored insulation to identify wiring based on signal characteristics. For example, you may use blue for dc wiring and red for ac wiring.

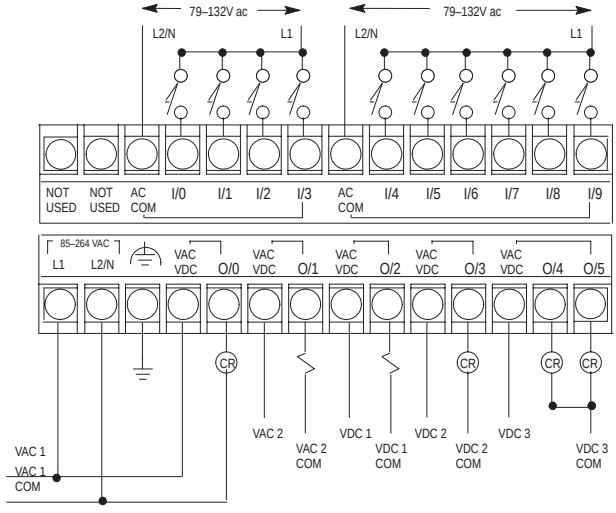
Wiring Diagrams, Discrete Input and Output Voltage Ranges

The following pages show the wiring diagrams, discrete input voltage ranges, and discrete output voltage ranges. Controllers with dc inputs can be wired as either sinking or sourcing configurations. (Sinking and sourcing does not apply to ac inputs.)

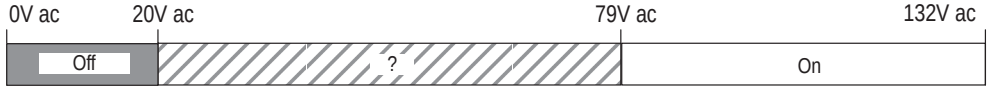
Note ⏚ *This symbol denotes a functional earth ground terminal which provides a low impedance path between electrical circuits and earth for non-safety purposes, such as noise immunity improvement.*

The 24V dc sensor power source should not be used to power output circuits. It should only be used to power input devices (e.g. sensors, switches). Refer to page 1–4 for information on MCR wiring in output circuits.

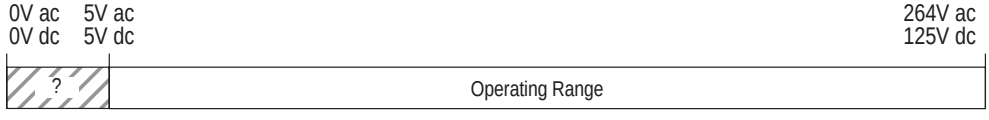
1761-L16AWA Wiring Diagram



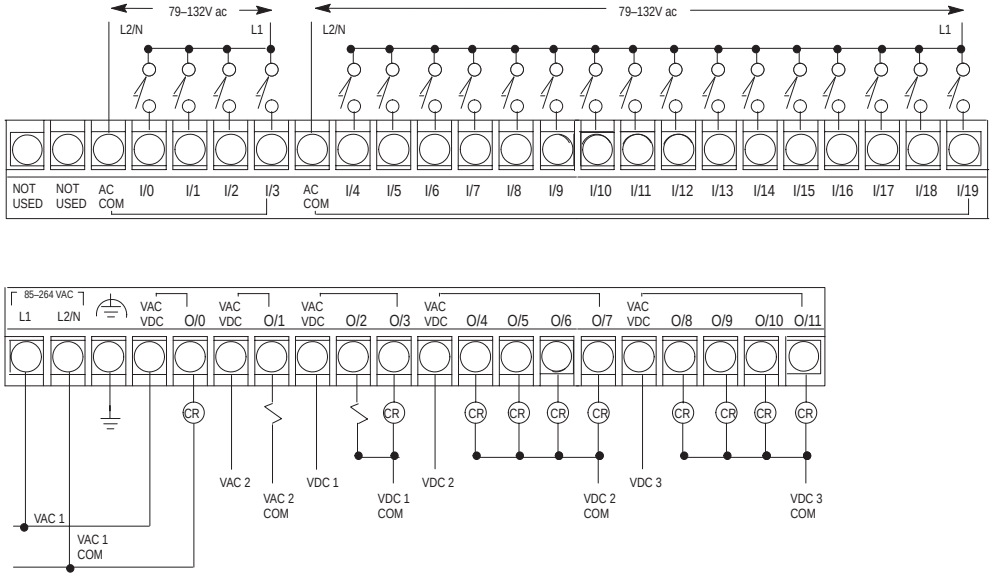
1761-L16AWA Input Voltage Range



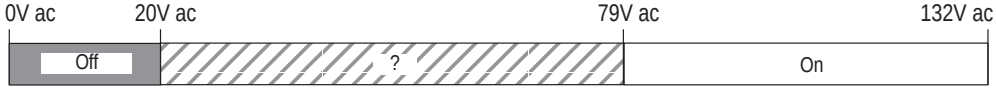
1761-L16AWA Output Voltage Range



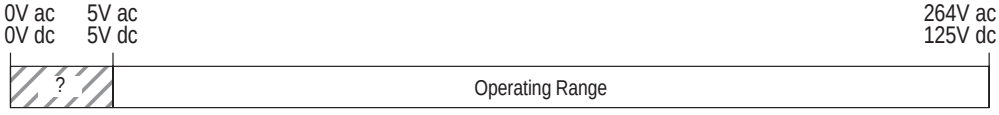
1761-L32AWA Wiring Diagram



1761-L32AWA Input Voltage Range

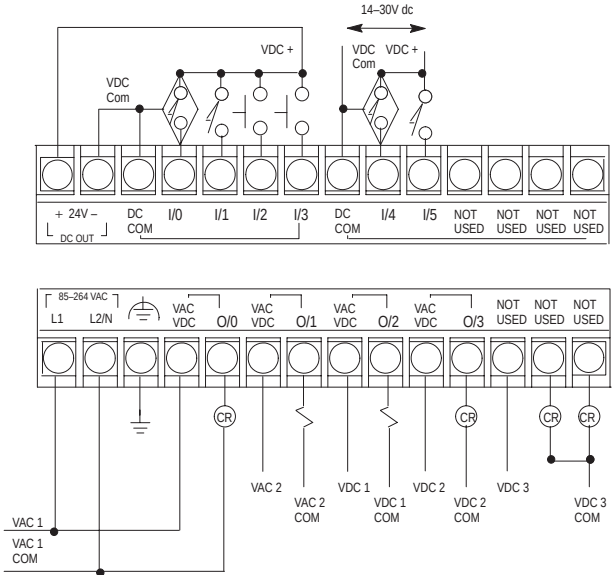


1761-L32AWA Output Voltage Range

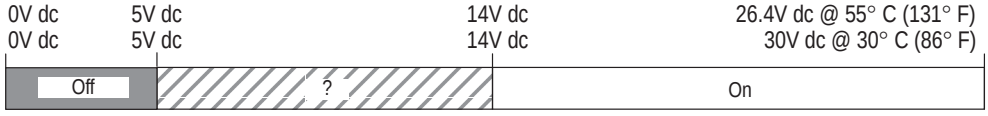


1761-L10BWA Wiring Diagram (Sinking Input Configuration)

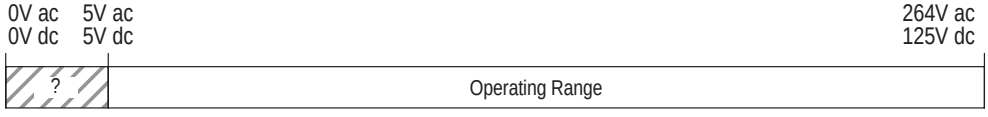
Note: Refer to page 2–3 for additional configuration options.



1761-L10BWA Input Voltage Range

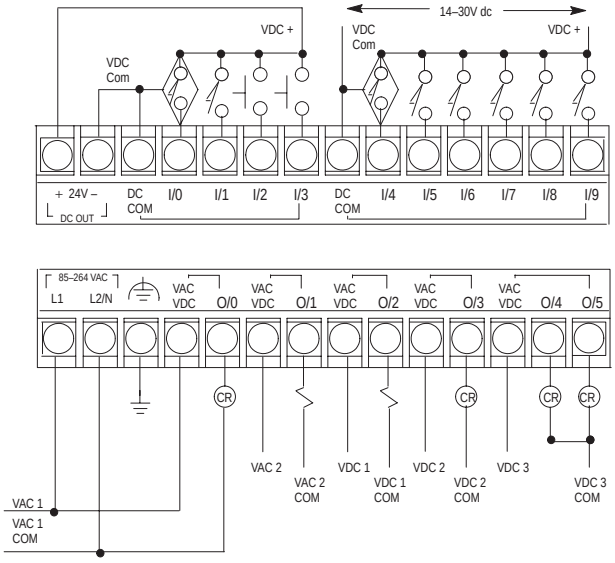


1761-L10BWA Output Voltage Range

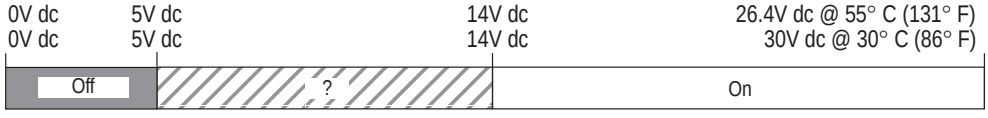


1761-L16BWA Wiring Diagrams (Sinking Input Configuration)

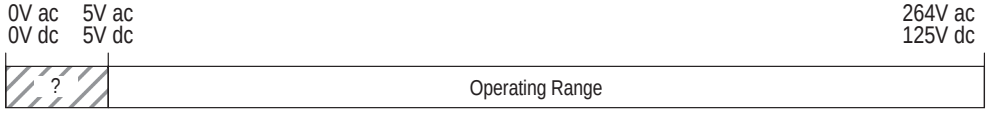
Note: Refer to page 2–3 for additional configuration options.



1761-L16BWA Input Voltage Range

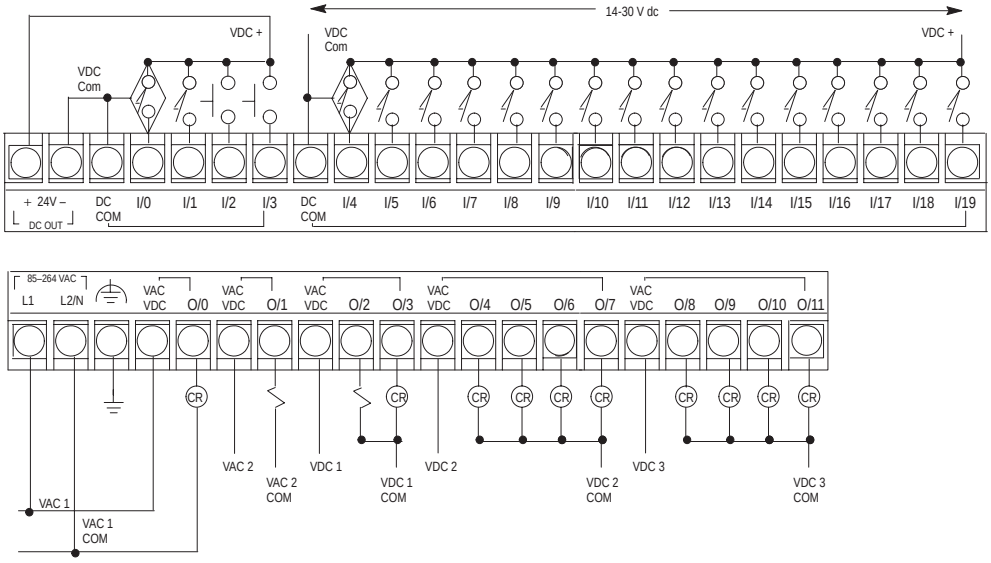


1761-L16BWA Output Voltage Range

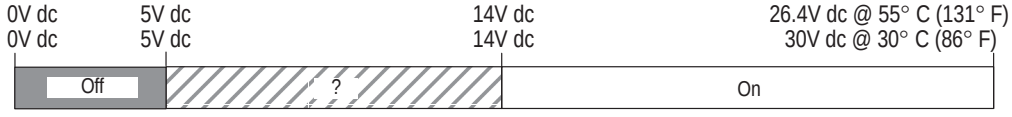


1761-L32BWA Wiring Diagram (Sinking Input Configuration)

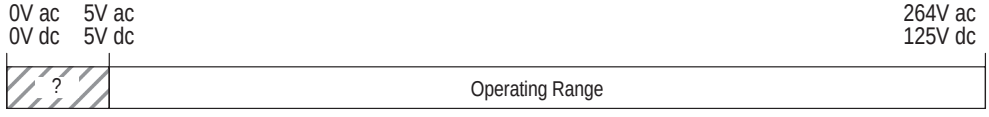
Note: Refer to page 2–3 for additional configuration options.



1761-L32BWA Input Voltage Range

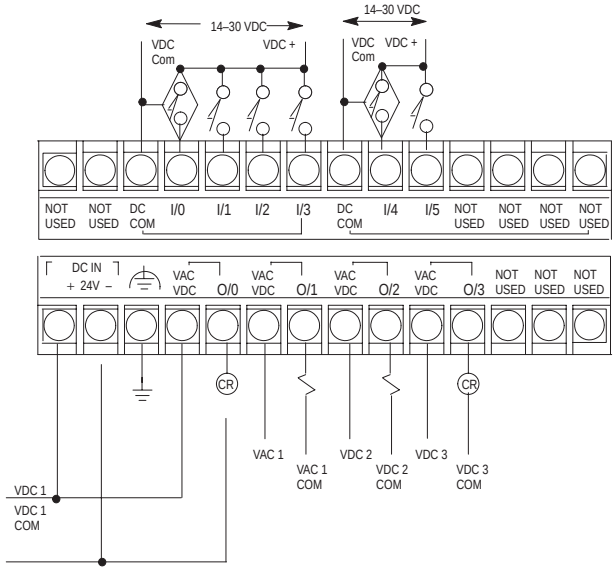


1761-L32BWA Output Voltage Range

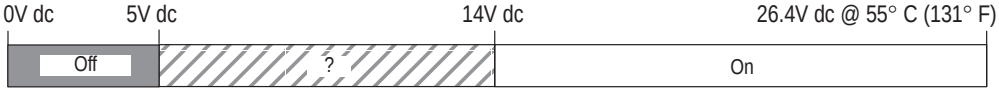


1761-L10BWB Wiring Diagram (Sinking Input Configuration)

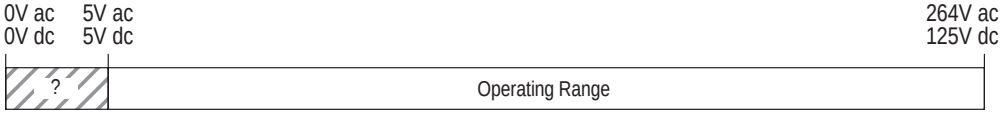
Note: Refer to page 2–4 for additional configuration options.



1761-L10BWB Input Voltage Range

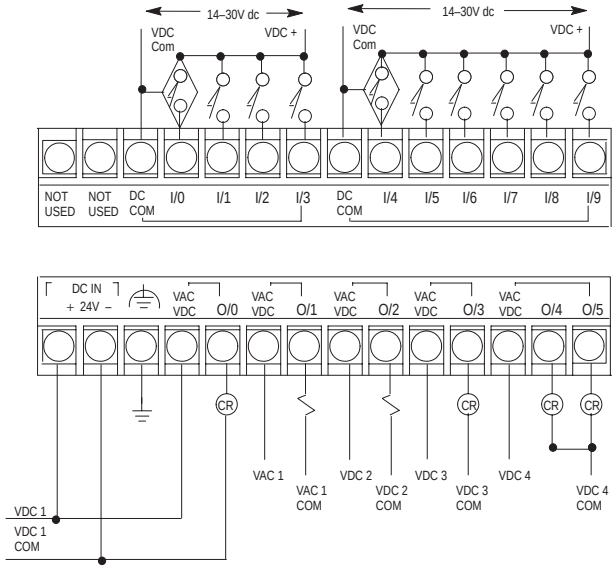


1761-L10BWB Output Voltage Range

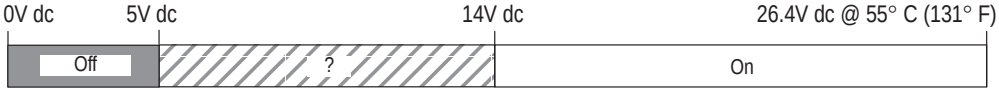


1761-L16BWB Wiring Diagram (Sinking Input Configuration)

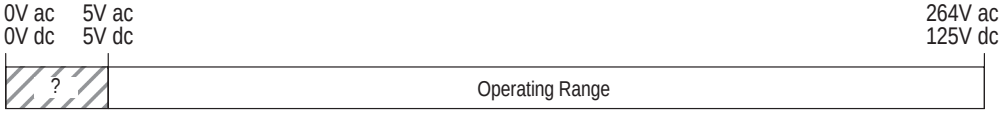
Note: Refer to page 2–4 for additional configuration options.



1761-L16BWB Input Voltage Range

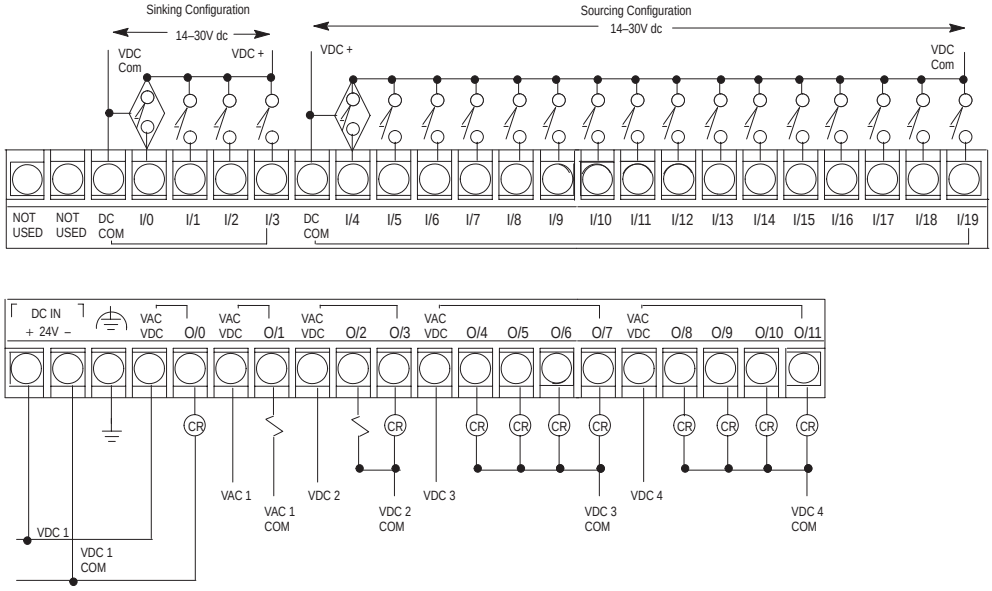


1761-L16BWB Output Voltage Range

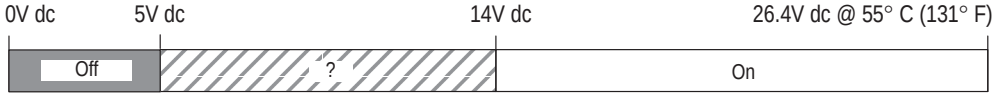


1761-L32BWB Wiring Diagram (Sinking Input Configuration)

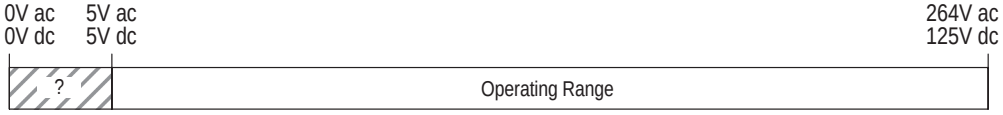
Note: Refer to page 2–4 for additional configuration options.



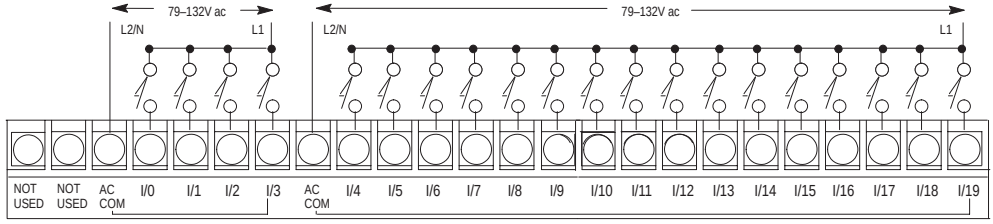
1761-L32BWB Input Voltage Range

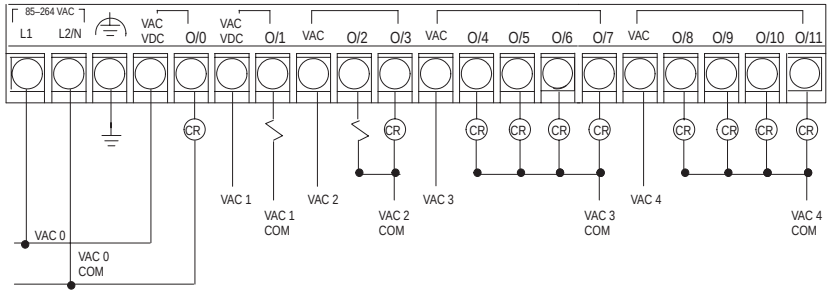


1761-L32BWB Output Voltage Range

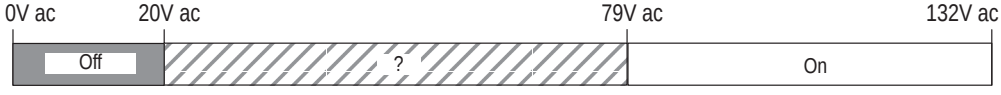


1761-L32AAA Wiring Diagram

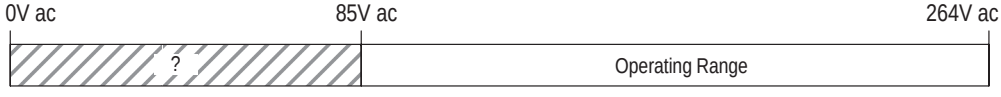




1761-L32AAA Input Voltage Range

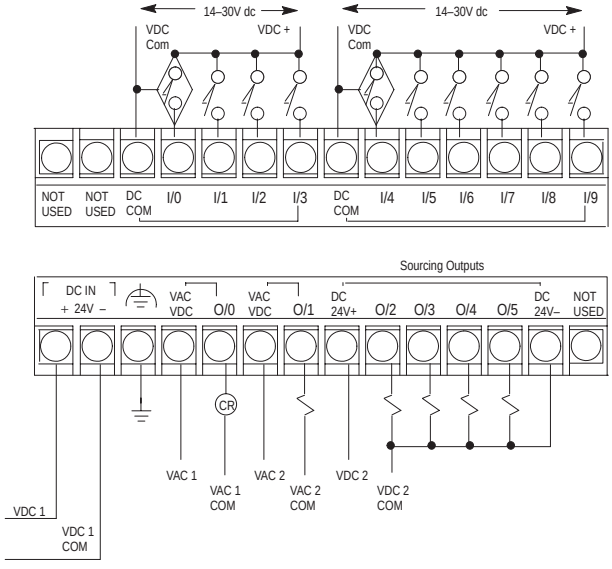


1761-L32AAA Output Voltage Range

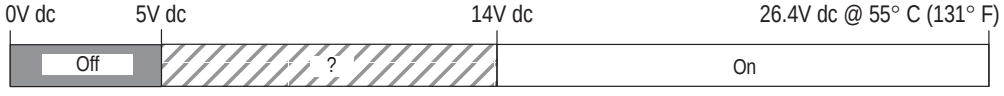


1761-L16BBB Wiring Diagrams (Sinking Input Configuration)

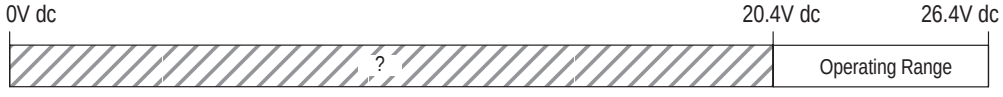
Note: Refer to page 2–4 for additional configuration options.



1761-L16BBB Input Voltage Range

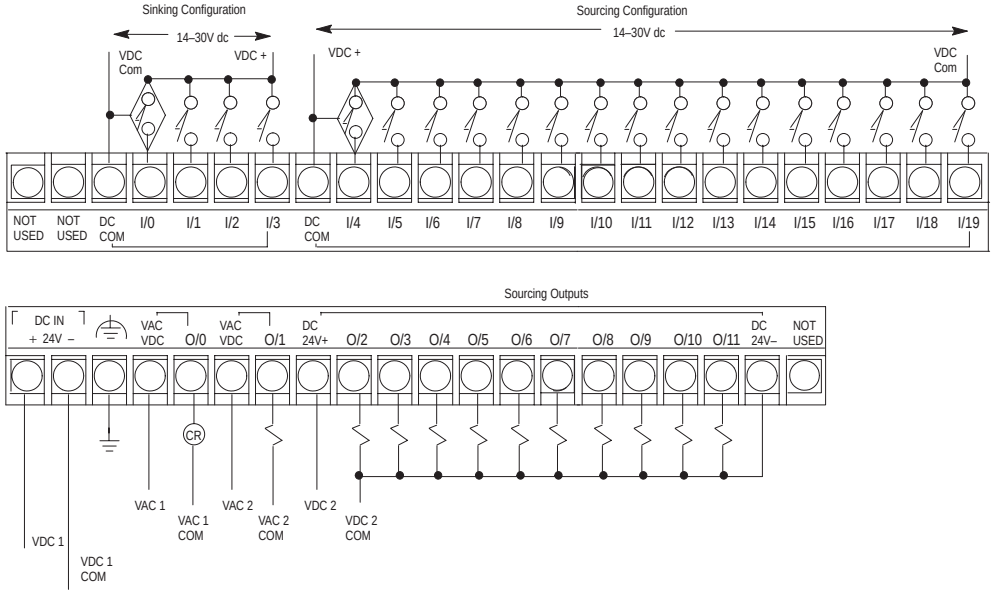


1761-L16BBB Output Voltage Range

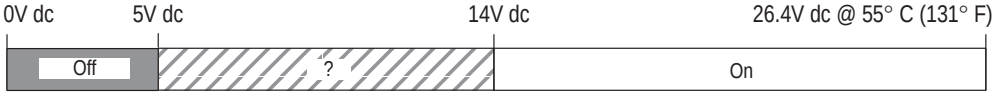


1761-L32BBB Wiring Diagram (Sinking Input Configuration)

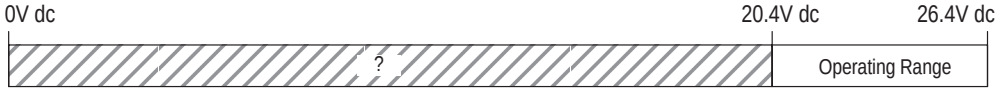
Note: Refer to page 2–4 for additional configuration options.



1761-L32BBB Input Voltage Range

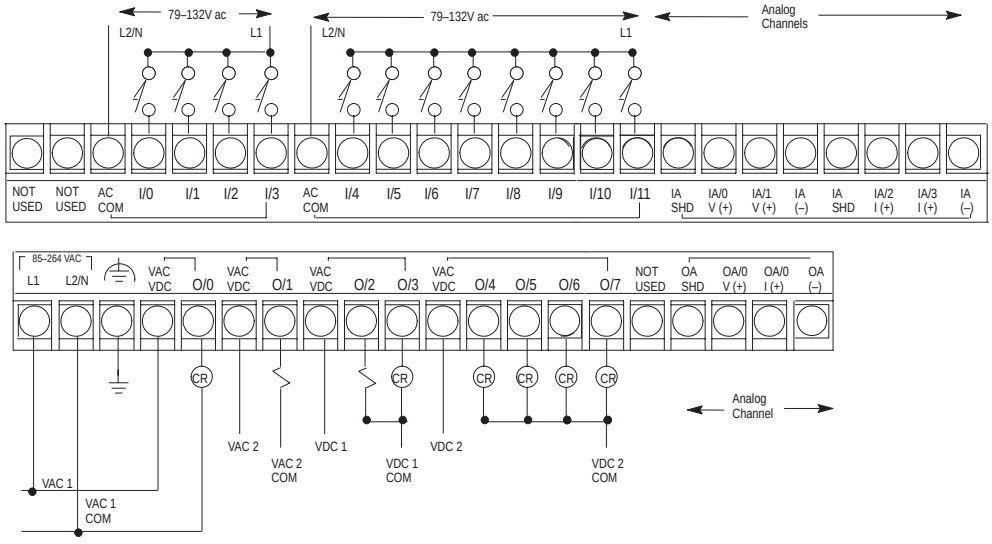


1761-L32BBB Output Voltage Range

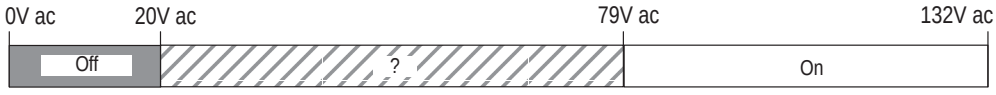


1761-L20AWA-5A Wiring Diagram

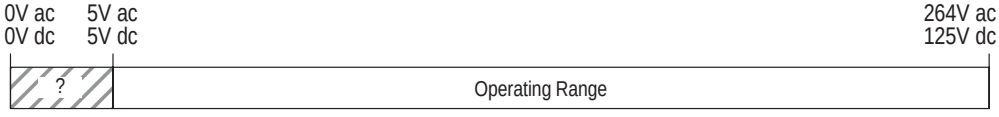
Note: Refer to pages 2–21 through 2–23 for additional information on analog wiring.



1761-L20AWA-5A Input Voltage Range



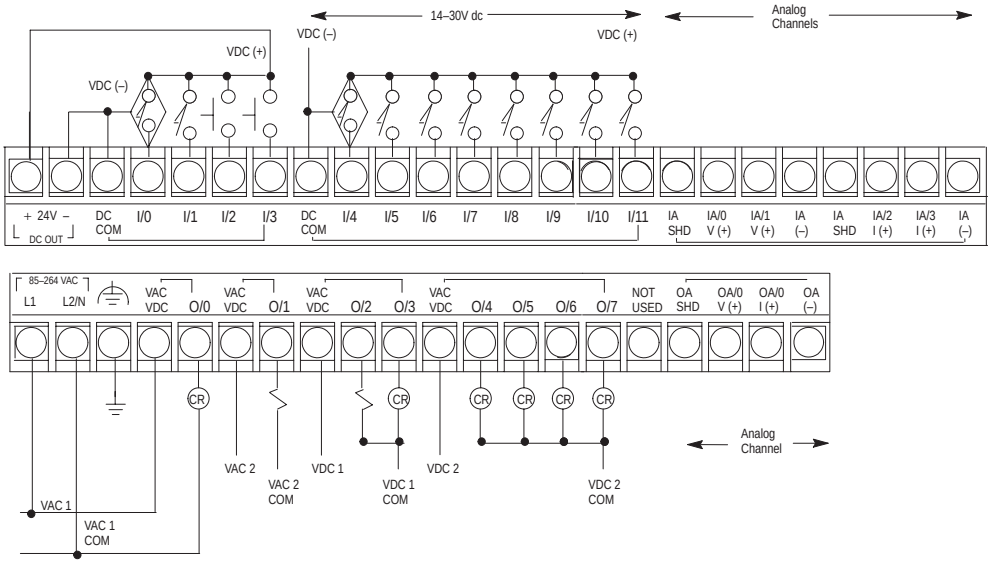
1761-L20AWA-5A Output Voltage Range



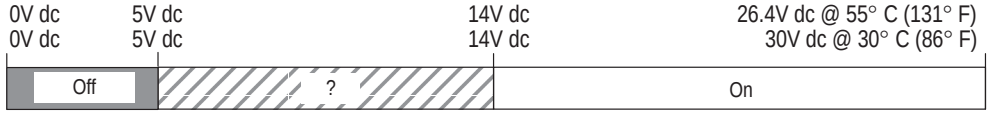
1761-L20BWA-5A Wiring Diagram (Sinking Input Configuration)

Note: Refer to page 2–3 for additional discrete configuration options.

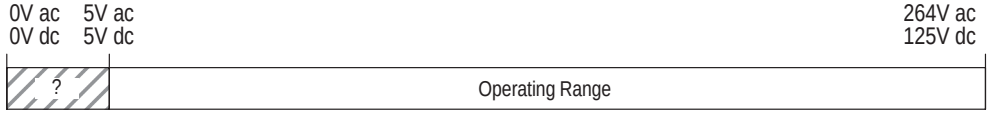
Refer to pages 2–21 through 2–23 for additional information on analog wiring.



1761-L20BWA-5A Discrete Input Voltage Range



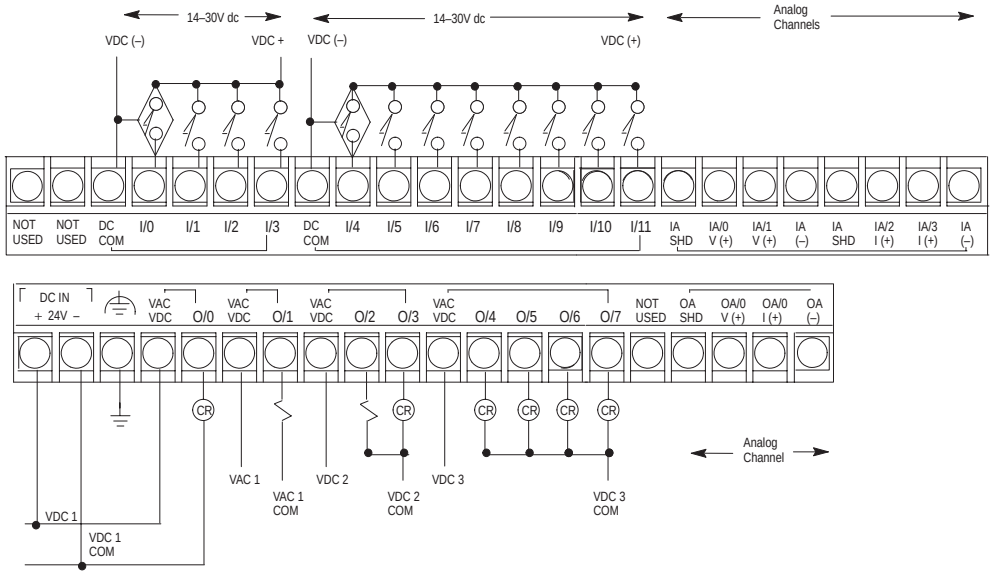
1761-L20BWA-5A Relay Output Voltage Range



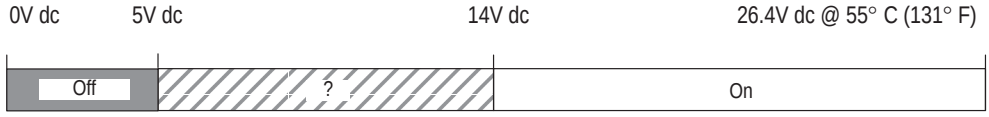
1761-L20BWB-5A Wiring Diagram (Sinking Input Configuration)

Note: Refer to page 2–4 for additional discrete configuration options.

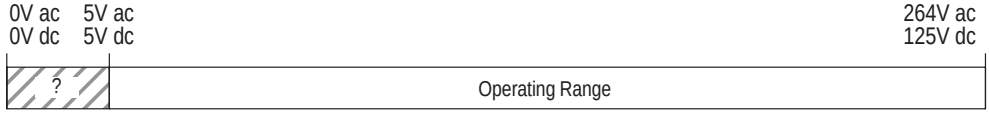
Refer to pages 2–21 through 2–23 for additional information on analog wiring.



1761-L20BWB-5A Discrete Input Voltage Range



1761-L20BWB-5A Relay Output Voltage Range



Minimizing Electrical Noise on Analog Controllers

Inputs on analog employ digital high frequency filters that significantly reduce the effects of electrical noise on input signals. However, because of the variety of applications and environments where analog controllers are installed and operating, it is impossible to ensure that all environmental noise will be removed by the input filters.

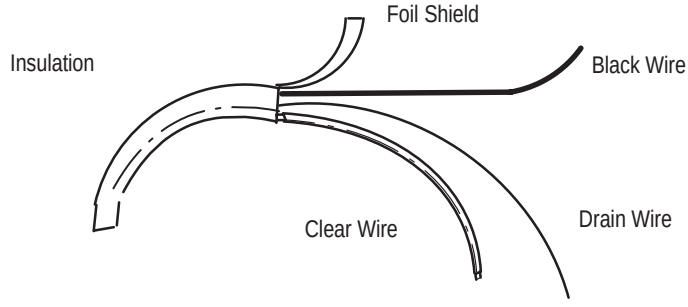
Several specific steps can be taken to help reduce the effects of environmental noise on analog signals:

- install the MicroLogix 1000 system in a properly rated (i.e., NEMA) enclosure. Make sure that the MicroLogix 1000 system is properly grounded.
- use Belden cable #8761 for wiring the analog channels making sure that the drain wire and foil shield are properly earth grounded at one end of the cable.
- route the Belden cable separate from any other wiring. Additional noise immunity can be obtained by routing the cables in grounded conduit.

A system may malfunction due to a change in the operating environment after a period of time. We recommend periodically checking system operation, particularly when new machinery or other noise sources are installed near the MicroLogix 1000 system.

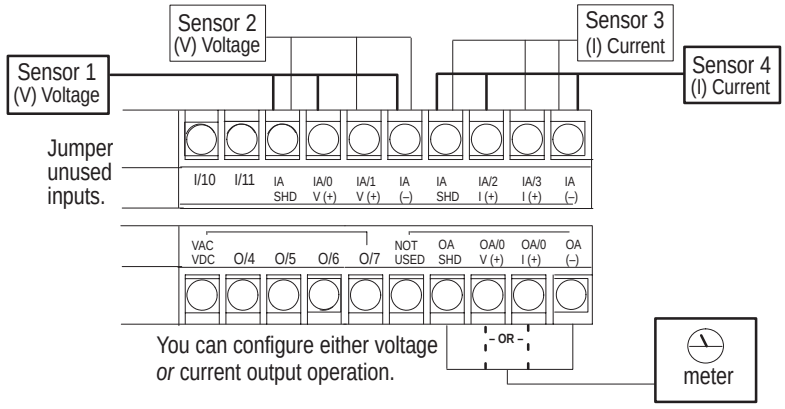
Grounding Your Analog Cable

Use shielded communication cable (Belden #8761). The Belden cable has two signal wires (black and clear), one drain wire and a foil shield. The drain wire and foil shield must be grounded at one end of the cable. *Do not* earth ground the drain wire and foil shield at *both* ends of the cable.



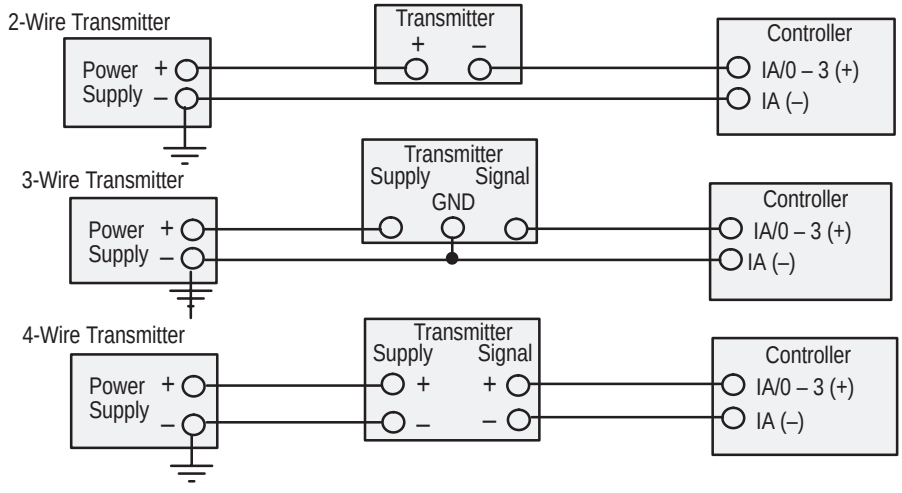
Wiring Your Analog Channels

Analog input circuits can monitor current *and* voltage signals and convert them to serial digital data. The analog output can support either a voltage *or* a current function.



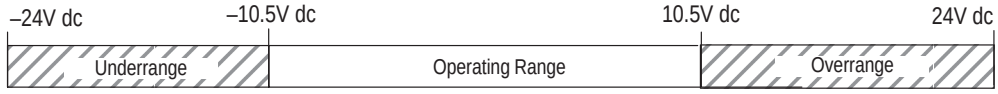
For increased noise immunity, connect a ground wire directly from the shield terminals to chassis ground.

Important: The controller does *not* provide loop power for analog inputs. Use a power supply that matches the transmitter specifications.



Analog Voltage and Current Input and Output Ranges

Analog Voltage Input Range



Analog Current Input Range

Note *The analog voltage inputs are protected to withstand the application of ±24V dc without damage to the controller. The analog current inputs are protected to withstand the application of ±50 mA without damage.*

Analog Voltage Output Range

Analog Current Output Range

Note *The analog outputs are protected to withstand the short circuiting of the voltage or current outputs without damage to the controller.*

For information on analog signal and data word values using the nominal transfer function formula, see page 5–5.

Wiring Your Controller for High-Speed Counter Applications

To wire the controller for high-speed counter applications use input terminals I/0, I/1, I/2, and I/3. Refer to chapter 12 for information on using the high-speed counter.

Shielded cable is required for high-speed input signals 0–3 when the filter setting is set to either 0.10 ms or 0.075 ms. We recommend Belden #9503 or equivalent for lengths up to 305 m (1000 ft). Shields should be grounded only at the signal source end of the cable. Ground the shield to the case of the signal source, so energy coupled to the shield will not be delivered to signal source's electronics.

3 Connecting the System

This chapter describes how to wire your controller system. The method you use and cabling required to connect your controller depends on what type of system you are employing. This chapter also describes how the controller establishes communication with the appropriate network.

For information on:	See page:
DF1 protocol connections	3–2
DH-485 network connections	3–5
Establishing communication	3–17

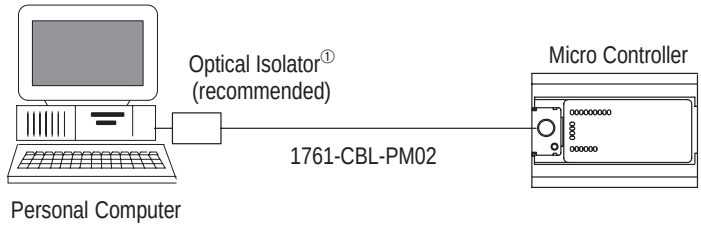
Connecting the DF1 Protocol

There are two ways to connect the MicroLogix 1000 programmable controller to your personal computer using the DF1 protocol: using an isolated point-to-point connection, or using a modem. Descriptions of these methods follow.

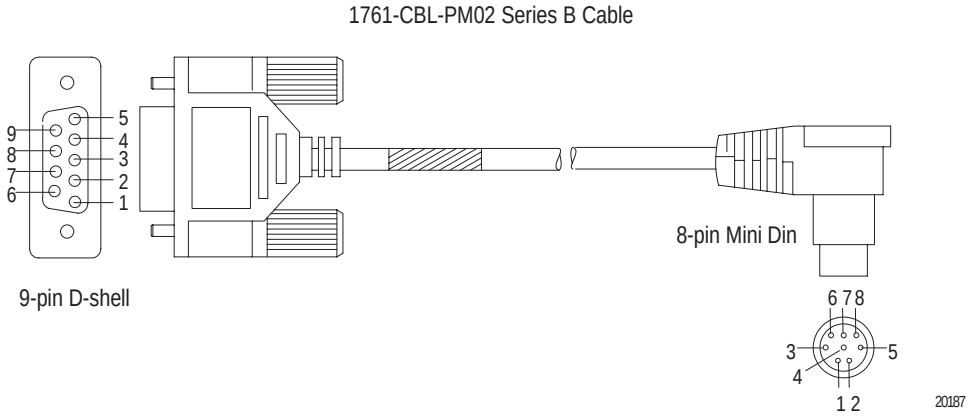
Chassis ground, user 24V ground, and RS-232 ground are internally connected. You must connect the chassis ground terminal screw to chassis ground prior to connecting any devices. It is important that you understand your personal computer's grounding system before connecting to the controller. An optical isolator is recommended between the controller and your personal computer.

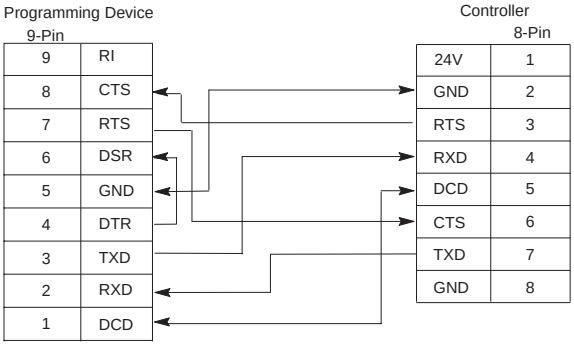
Making an Isolated Point-to-Point Connection

You can connect the MicroLogix 1000 programmable controller to your personal computer using a serial cable from your personal computer's serial port to the micro controller.



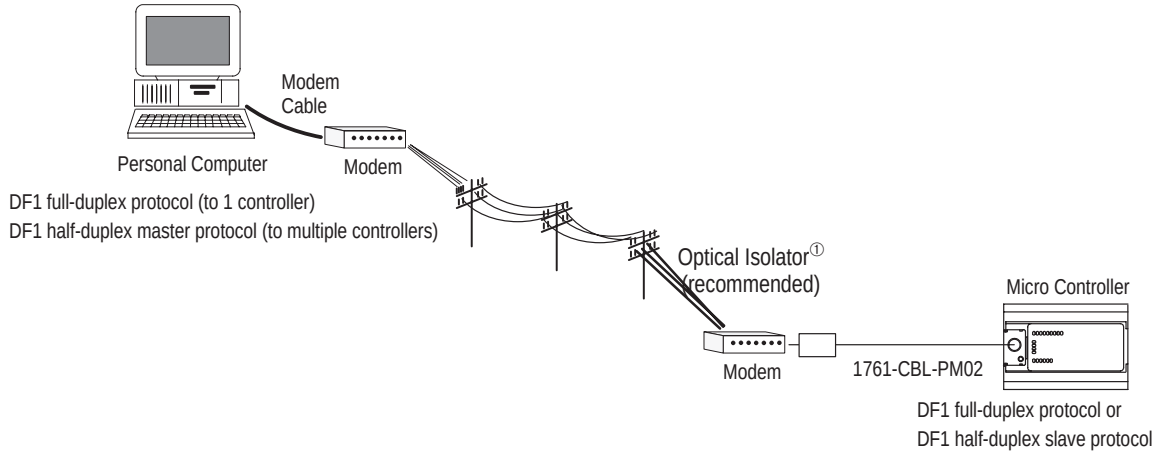
① We recommend using an AIC+, catalog number 1761-NET-AIC, as your optical isolator. See page 3–11 for specific AIC+ cabling information.

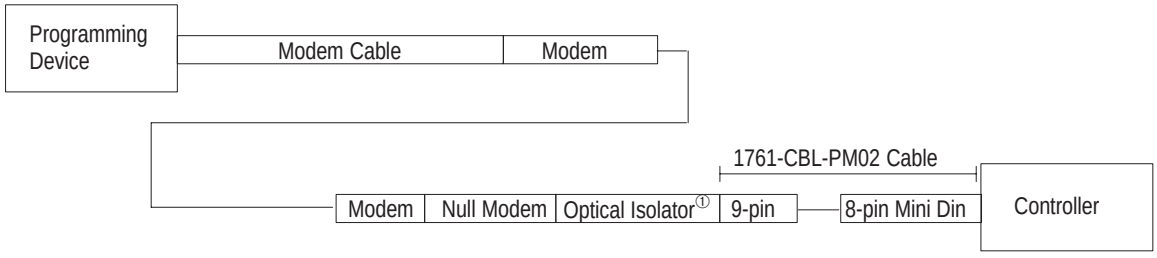




Using a Modem

You can also use modems to connect a personal computer to one MicroLogix 1000 controller (using DF1 full-duplex protocol) or to multiple controllers (using DF1 half-duplex protocol), as shown in the illustration that follows. Do not attempt to use DH-485 protocol through modems under any circumstance. (For information on types of modems you can use with the micro controllers, see page D–9.)

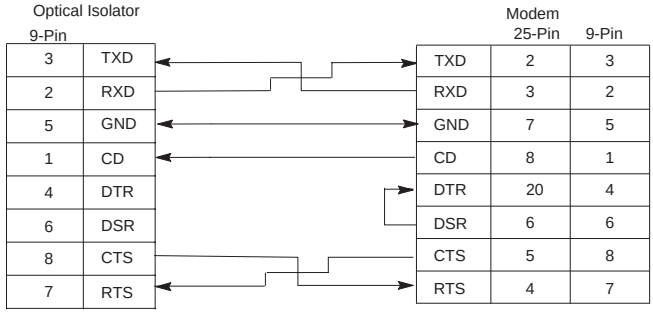




① We recommend using an AIC+, catalog number 1761-NET-AIC, as your optical isolator. See page 3–11 for specific AIC+ cabling information.

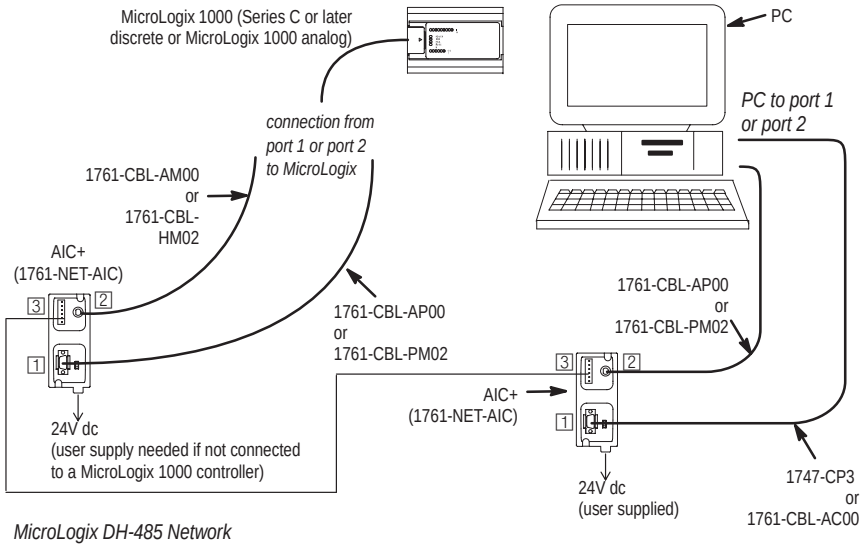
Constructing Your Own Null Modem Cable

If you construct your own null modem cable, the maximum cable length is 15.24 m (50 ft) with a 25-pin or 9-pin connector. Refer to the following typical pinout:



Connecting to a DH-485 Network

Note *Only Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers support DH-485 network connections.*


MicroLogix 1000 (Series C or later discrete or MicroLogix 1000 analog)
PC
PC to port 1 or port 2
connection from port 1 or port 2 to MicroLogix
1761-CBL-AM00 or 1761-CBL-HM02
AIC+ (1761-NET-AIC)
1761-CBL-AP00 or 1761-CBL-PM02
1761-CBL-AP00 or 1761-CBL-PM02
AIC+ (1761-NET-AIC)
24V dc (user supply needed if not connected to a MicroLogix 1000 controller)
24V dc (user supplied)
1747-CP3 or 1761-CBL-AC00

MicroLogix DH-485 Network

1 DB-9 RS-232 port
2 mini-DIN 8 RS-232 port
3 DH-485 port

Recommended Tools

To connect a DH-485 network, you need tools to strip the shielded cable and to attach the cable and terminators to the AIC+ Advanced Interface Converter. We recommend the following equipment (or equivalent):

Description	Part Number	Manufacturer
Shielded Twisted Pair Cable	#3106A or #9842	Belden
Stripping Tool	45-164	Ideal Industries
1/8 " Slotted Screwdriver	Not Applicable	Not Applicable

DH-485 Communication Cable

The suggested DH-485 communication cable is either Belden #3106A or #9842. The cable is jacketed and shielded with one or two twisted wire pairs and a drain wire.

One pair provides a balanced signal line, and one additional wire is used for a common reference line between all nodes on the network. The shield reduces the effect of electrostatic noise from the industrial environment on network communication.

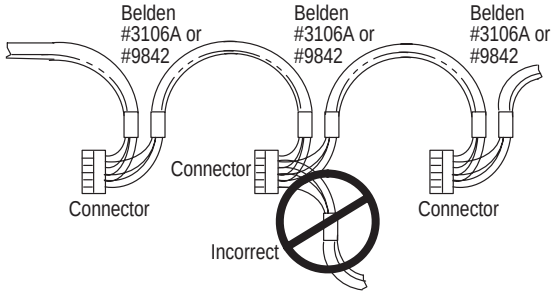
The communication cable consists of a number of cable segments daisy-chained together. The total length of the cable segments cannot exceed 1219 m (4000 ft).

When cutting cable segments, make them long enough to route them from one AIC+ to the next with sufficient slack to prevent strain on the connector. Allow enough extra cable to prevent chafing and kinking in the cable.

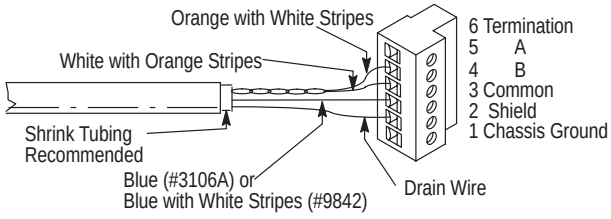
Use these instructions for wiring the Belden #3106A or #9842 cable. (If you are using standard Allen-Bradley cables, see the Cable Selection Guide on page 3–12.)

Connecting the Communication Cable to the DH-485 Connector

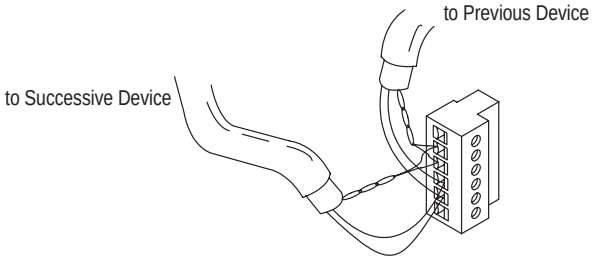
Note *A daisy-chained network is recommended. We do **not** recommend the following:*



Single Cable Connection



Multiple Cable Connection



The table below shows connections for Belden #3106A.

For this Wire/Pair	Connect this Wire	To this Terminal
Shield/Drain	Non-jacketed	Terminal 2 – Shield
Blue	Blue	Terminal 3 – (Common)
White/Orange	White with Orange Stripe	Terminal 4 – (Data B)
	Orange with White Stripe	Terminal 5 – (Data A)

The table below shows connections for Belden #9842.

For this Wire/Pair	Connect this Wire	To this Terminal
Shield/Drain	Non-jacketed	Terminal 2 – Shield
Blue/White	White with Blue Stripe	Cut back – no connection①
	Blue with White Stripe	Terminal 3 – (Common)
White/Orange	White with Orange Stripe	Terminal 4 – (Data B)
	Orange with White Stripe	Terminal 5 – (Data A)

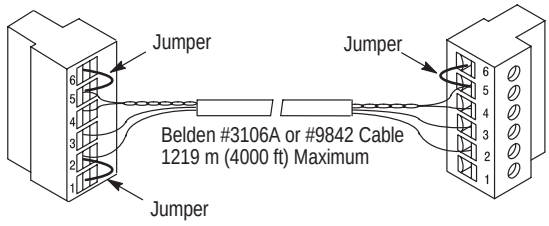
① To prevent confusion when installing the communication cable, cut back the white with blue stripe wire immediately after the the insulation jacket is removed. This wire is not used by DH-485.

Grounding and Terminating the DH-485 Network

Only one connector at the end of the link must have Terminals 1 and 2 jumpered together. This provides an earth ground connection for the shield of the communication cable.

Both ends of the network must have Terminals 5 and 6 jumpered together. This connects the termination impedance (of 120Ω) that is built into each AIC+ as required by the DH-485 specification.

End-of-Line Termination



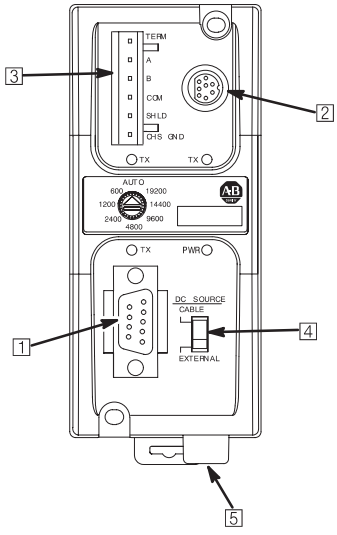
Connecting the AIC+

Note *Only Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers support DH-485 connections with the AIC+.*

You can connect an unpowered AIC+, catalog number 1761-NET-AIC, to the network without disrupting network activity. In addition, if a MicroLogix 1000 controller powers an AIC+ that is connected to the network, network activity will not be disrupted should the MicroLogix 1000 controller be removed from the AIC+.

The figure that follows shows the external wiring connections and specifications of the AIC+.

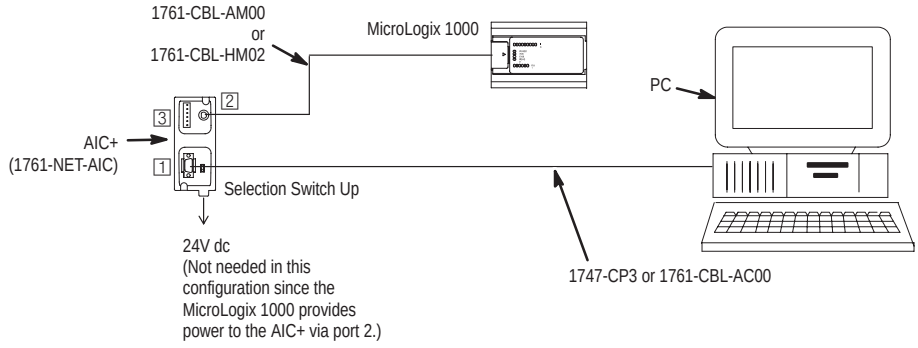
AIC+ Advanced Interface Converter (1761-NET-AIC)



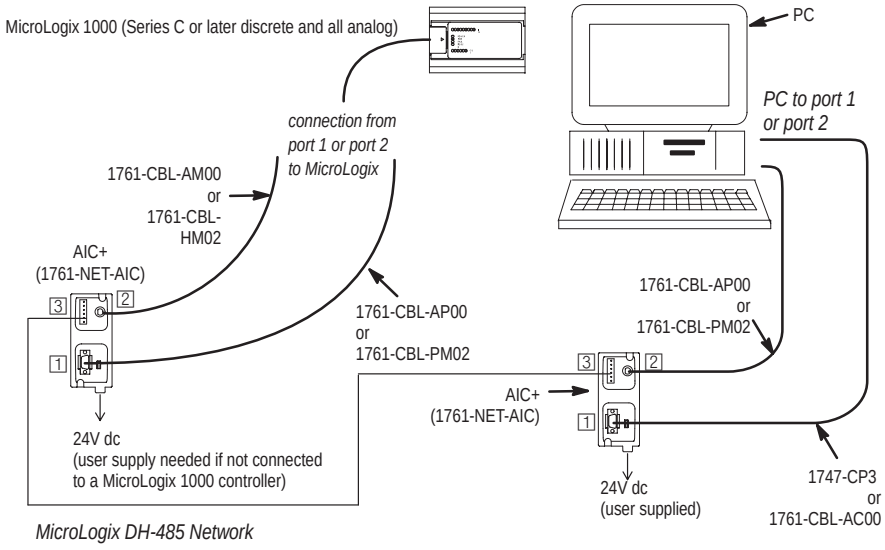
Item	Description
1	Port 1 – DB-9 RS-232, DTE
2	Port 2 – mini-DIN 8 RS-232
3	Port 3 – DH-485 Phoenix plug
4	DC Power Source selector switch (cable = port 2 power source, external = external power source connected to item 5)
5	Terminals for external 24V dc power supply and chassis ground

For additional information on connecting the AIC+, see the *Advanced Interface Converter (AIC+) and DeviceNet Interface (DNI) Installation Instructions*, Publication 1761-5.11.

DF1 Isolated Point-to-Point Connection



DH-485 Network Connection



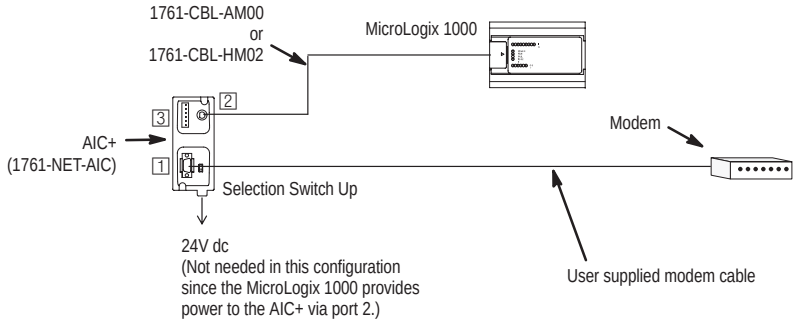
MicroLogix DH-485 Network

1 DB-9 RS-232 port

2 mini-DIN 8 RS-232 port

3 DH-485 port

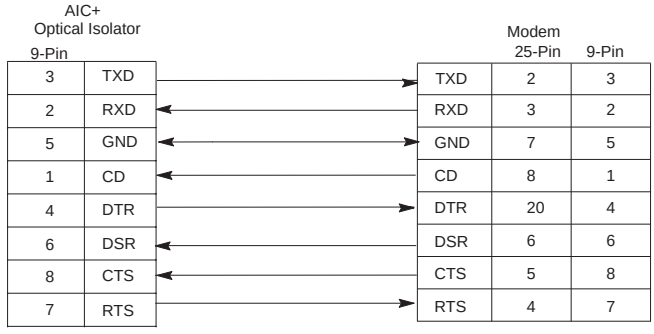
DF1 Isolated Modem Connection



For additional information on connections using the AIC+, see the *Advanced Interface Converter (AIC+) and DeviceNet Interface (DNI) Installation Instructions*, Publication 1761-5.11.

Constructing Your Own Modem Cable

If you construct your own modem cable, the maximum cable length is 15.24 m (50 ft) with a 25-pin or 9-pin connector. Refer to the following typical pinout:



Cable Selection Guide

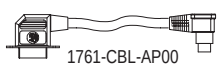
Cable	Length	Connections from	to AIC+	External Power Supply Required②	Power Selection Switch Setting②
1747-CP3 1761-CBL-AC00①	3m (9.8 ft) 45 cm (17.7 in)	SLC 5/03 or SLC 5/04 processor, channel 0	port 1	yes	external
		PC COM port	port 1	yes	external
		PanelView 550 through NULL modem adapter	port 1	yes	external
		DTAM Plus / DTAM Micro™	port 1	yes	external
		Port 1 on another AIC+	port 1	yes	external

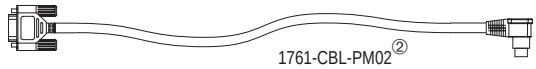
Cable	Length	Connections from	to AIC+	External Power Supply Required①	Power Selection Switch Setting①
1761-CBL-AS03 1761-CBL-AS09	3m (9.8 ft) 9.5m (31.17 ft)	SLC 500 Fixed, SLC 5/01, SLC 5/02, and SLC 5/03 processors	port 3	yes	external
		PanelView 550 RJ45 port	port 3	yes	external

Cable	Length	Connections from	to AIC+	External Power Supply Required	Power Selection Switch Setting
1761-CBL-AM00	45 cm (17.7 in)	MicroLogix 1000	port 2	no	cable
1761-CBL-HM02②	2m (6.5 ft)	to port 2 on another AIC+	port 2	yes	external

① External power supply required unless the AIC+ is powered by the device connected to port 2, then the selection switch should be set to cable.

② Series B or higher cables are required for hardware handshaking.





Cable	Length	Connections from	to AIC+	External Power Supply Required	Power Selection Switch Setting
1761-CBL-AP00 1761-CBL-PM02②	45 cm (17.7 in) 2m (6.5 ft)	SLC 5/03 or SLC 5/04 processors, channel 0	port 2	yes	external
		MicroLogix 1000	port 1	yes①	external①
		PanelView 550 through NULL modem adapter	port 2	yes	external
		DTAM Plus / DTAM Micro	port 2	yes	external
		PC COM port	port 2	yes	external

Cable	Length	Connections from	to AIC+	External Power Supply Required	Power Selection Switch Setting
straight 9–25 pin	—	modem or other communication device	port 1	yes①	external①

① External power supply required unless the AIC+ is powered by the device connected to port 2, then the selection switch should be set to cable.

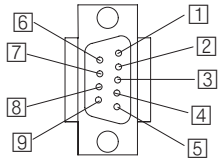
② Series B or higher cables are required for hardware handshaking.

Recommended User-Supplied Components

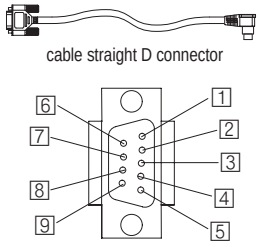
These components can be purchased from your local electronics supplier.

Component	Recommended Model
external power supply and chassis ground	power supply rated for 20.4–28.8V dc
NULL modem adapter	standard AT
straight 9–25 pin RS-232 cable	see table below for port information if making own cables

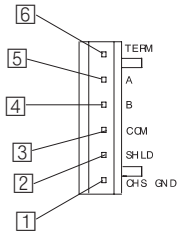
DB-9 RS-232 Port 1



1761-CBL-AP00 or 1761-CBL-PM02



DH-485 connector Port 3



Pin	Port 1 DB-9 RS-232	Port 2[①] (1761-CBL-PM02 cable)	Port 3 DH-485 Connector
1	received line signal detector (DCD)	same state as port 1's DCD signal	chassis ground
2	received data (RxD)	received data (RxD)	cable shield
3	transmitted data (TxD)	transmitted data (TxD)	signal ground
4	DTE ready (DTR)[②]	DTE ready (DTR)[③]	DH-485 data B
5	signal common (GRD)	signal common (GRD)	DH-485 data A
6	DCE ready (DSR)[②]	DCE ready (DSR)[③]	termination
7	request to send (RTS)	request to send (RTS)	not applicable
8	clear to send (CTS)	clear to send (CTS)	not applicable
9	not applicable	not applicable	not applicable

① An 8-pin mini DIN connector is used for making connections to port 2. This connector is not commercially available. If you are making a cable to connect to port 2, you must configure your cable to connect to the Allen-Bradley cable shown above.

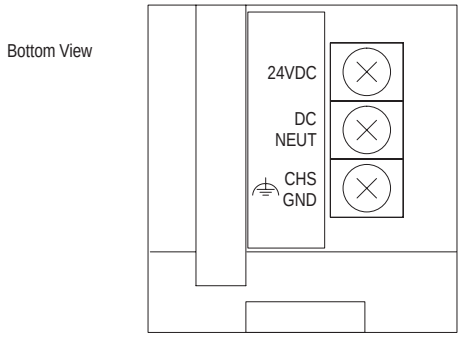
② On port 1, pin 4 is electronically jumpered to pin 6. Whenever the AIC+ is powered on, pin 4 will match the state of pin 6.

③ In the 1761-CBL-PM02 cable, pins 4 and 6 are jumpered together within the DB-9 connector.

Powering the AIC+

If you use an external power supply, it must be 24V dc. Permanent damage will result if miswired with the wrong power source.

Set the DC Power Source selector switch to EXTERNAL before connecting the power supply to the AIC+.



Always connect the CHS GND (chassis ground) terminal to the nearest earth ground. This connection must be made whether or not an external 24V dc supply is used.

In normal operation with the MicroLogix 1000 programmable controller connected to port 2 of the AIC+, the controller powers the AIC+. Any AIC+ not connected to a controller requires a 24V dc power supply. The AIC+ requires 104 mA at 24V dc.

If both the controller and external power are connected to the AIC+, the power selection switch determines what device powers the AIC+.

Power Options

Below are two options for powering the AIC+:

- Use the 24V dc user power supply (200 mA maximum) built into the MicroLogix controller. The AIC+ is powered through a hard-wired connection using a communication cable (1761-CBL-HM02, or equivalent) connected to port 2.
- Use an external DC power supply with the following specifications:
  - operating voltage: 24V dc +20% / –15%
  - output current: 200 mA maximum
  - rated NEC

  Make a hard-wired connection from the external supply to the screw terminals on the bottom of the AIC+.

If you use an external power supply, it must be 24V dc. Permanent damage will result if miswired with the wrong power source.

Installing and Attaching the AIC+

1. Take care when installing the AIC+ in an enclosure so that the cable connecting the MicroLogix 1000 controller to the AIC+ does not interfere with the enclosure door.

2. Carefully plug the terminal block into the DH-485 port on the AIC+ you are putting on the network. Allow enough cable slack to prevent stress on the plug.

3. Provide strain relief for the Belden cable after it is wired to the terminal block. This guards against breakage of the Belden cable wires.

Establishing Communication

When you connect a MicroLogix 1000 controller to a network, it *automatically* finds which protocol is active (DF1 or DH-485), and establishes communication accordingly. Therefore, no special configuration is required to connect to either network.

However, to shorten the connection time, you can specify which protocol the controller should attempt to establish communication with first. This is done using the Primary Protocol bit, S:0/10. The default setting for this bit is DF1 (0). If the primary protocol bit is set to DF1, the MicroLogix 1000 controller will attempt to connect using the configured DF1 protocol; either full-duplex or half-duplex slave. To have the controller first attempt DH-485 communication, set this bit to 1.

For DH-485 networks that will only contain MicroLogix controllers, at least one controller must have its primary protocol bit set to 1 so that the network can be initialized.

Automatic Protocol Switching

The MicroLogix 1000 Series D or later discrete and all MicroLogix 1000 analog controllers perform automatic protocol switching between DH-485 and the configured DF1 protocol. (The controller cannot automatically switch between DF1 full-duplex and DF1 half-duplex slave.) This feature allows you to switch from active communication on a DF1 half-duplex network to the DH-485 protocol to make program changes.

Simply disconnect the MicroLogix controller from the DF1 half-duplex network and connect it to your personal computer. The controller recognizes the computer is attempting to communicate using the DH-485 protocol and automatically switches to it. When your program changes are complete, you can disconnect your computer, reconnect the modem, and the controller automatically switches back to the configured DF1 protocol. For example, if you are using the DH-485 protocol to make program changes and you connect an HHP, you can switch to active communication on a DF1 full-duplex network.

The following baud rate limitations affect autoswitching:

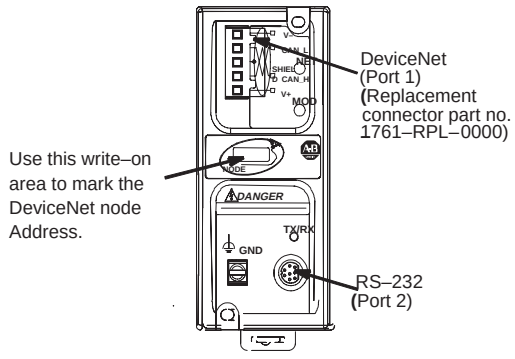
- If the configured DH-485 baud rate is 19200, the configured DF1 baud rate must be 4800 or greater.
- If the configured DH-485 baud rate is 9600, the configured DF1 baud rate must be 2400 or greater.

DeviceNet Communications

You can also connect a MicroLogix to a DeviceNet network using the DeviceNet Interface (DNI), catalog number 1761-NET-DNI. For additional information on connecting the DNI, see the *Advanced Interface Converter (AIC+) and DeviceNet Interface (DNI) Installation Instructions*, Publication 1761-5.11. For information on how to configure and commission a DNI, see the *DeviceNet Interface User Manual*, Publication 1761-6.5.

The figure that follows identifies the ports of the DNI.

DNI DeviceNet Interface (1761-NET-DNI)



Cable Selection Guide

Cable	Length	Connections from	to DNI
1761-CBL-AM00	45 cm (17.7 in)	MicroLogix 1000 (all series)	port 2
1761-CBL-HM02①	2m (6.5 ft)	MicroLogix 1000 (all series)	port 2

Cable	Length	Connections from	to DNI
1761-CBL-APM00	45 cm (17.7 in)	SLC 5/03 or SLC 5/04 processors, channel 0	port 2
1761-CBL-PM02①	2m (6.5 ft)	PC COM port	port 2

① Series B cables or higher are required for hardware handshaking.

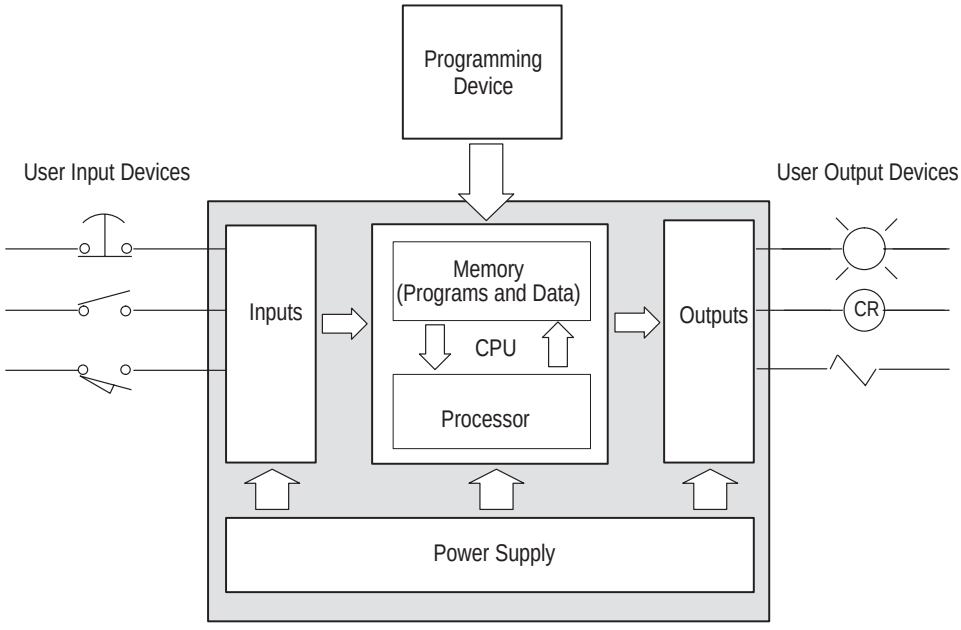
4 Programming Overview

This chapter explains how to program the MicroLogix 1000 programmable controller. Read this chapter for basic information about:

- principles of machine control
- understanding file organization and addressing
- understanding how processor files are stored and accessed
- applying ladder logic to your schematics
- a model for developing your program

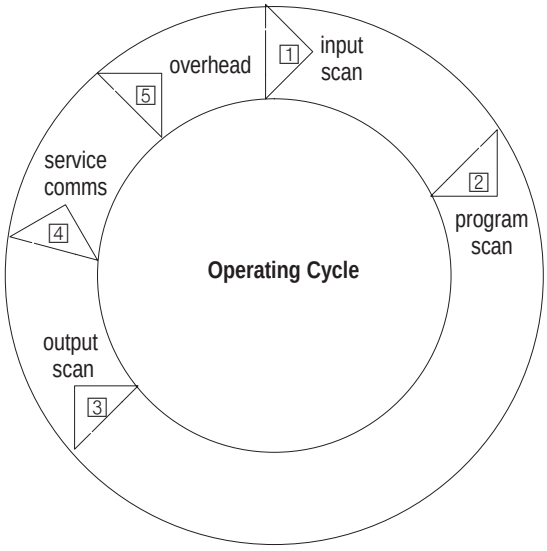
Principles of Machine Control

The controller consists of a built-in power supply, central processing unit (CPU), inputs, which you wire to input devices (such as pushbuttons, proximity sensors, limit switches), and outputs, which you wire to output devices (such as motor starters, solid-state relays, and indicator lights).



MicroLogix 1000 Programmable Controller

With the logic program entered into the controller, placing the controller in the Run mode initiates an operating cycle. The controller's operating cycle consists of a series of operations performed sequentially and repeatedly, unless altered by your program logic.



1 input scan – the time required for the controller to scan and read all input data; typically accomplished within μseconds.

2 program scan – the time required for the processor to execute the instructions in the program. The program scan time varies depending on the instructions used and each instruction's status during the scan time.

Note *Subroutine and interrupt instructions within your logic program may cause deviations in the way the operating cycle is sequenced.*

3 output scan – the time required for the controller to scan and write all output data; typically accomplished within μseconds.

4 service communications – the part of the operating cycle in which communication takes place with other devices, such as an HHP or personal computer.

5 housekeeping and overhead – time spent on memory management and updating timers and internal registers.

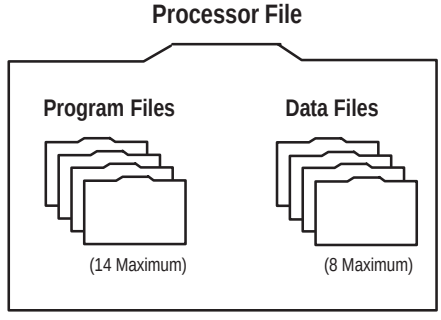
You enter a logic program into the controller using a programming device. The logic program is based on your electrical relay print diagrams. It contains instructions that direct control of your application.

Understanding File Organization

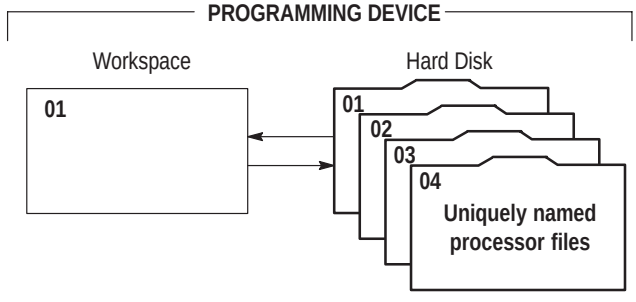
The processor provides control through the use of a program you create, called a processor file. This file contains other files that break your program down into more manageable parts.

Processor File Overview

Most of the operations you perform with the programming device involve the processor file and the two components created with it: program files and data files.



The programming device stores processor files on *hard disk (or floppy disk)*. Monitoring and editing of processor files is done in the *workspace* of the computer. After you select a file from disk and edit it, you then *save* the file hard to disk, replacing the original disk version with the edited version. The hard disk is the recommended location for a processor file.



Processor files are created in the offline mode using the programming device. These files are then restored (downloaded), to the processor for online operation.

Program Files

Program files contain controller information, the main ladder program, interrupt subroutines, and any subroutine programs. These files are:

- **System Program** (file 0) – This file contains various system related information and user-programmed information such as processor type, I/O configuration, processor file name, and password.
- **Reserved** (file 1) – This file is reserved.
- **Main Ladder Program** (file 2) – This file contains user-programmed instructions defining how the controller is to operate.
- **User Error Fault Routine** (file 3) – This file is executed when a recoverable fault occurs.
- **High-Speed Counter Interrupt** (file 4) – This file is executed when an HSC interrupt occurs. It can also be used for a subroutine ladder program.
- **Selectable Timed Interrupt** (file 5) – This file is executed when an STI occurs. It can also be used for a subroutine ladder program.
- **Subroutine Ladder Program** (files 6 – 15) – These are used according to subroutine instructions residing in the main ladder program file or other subroutine files.

Data Files

Data files contain the status information associated with external I/O and all other instructions you use in your main and subroutine ladder program files. In addition, these files store information concerning processor operation. You can also use the files to store “recipes” and look-up tables if needed.

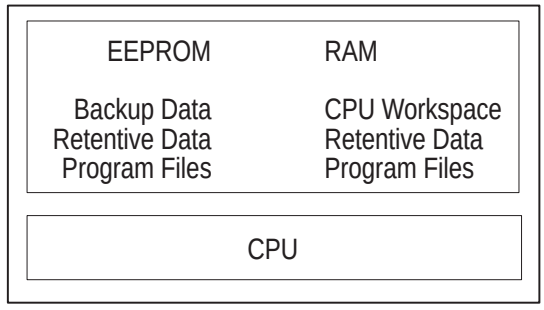
These files are organized by the type of data they contain. The data file types are:

- **Output** (file 0) – This file stores the state of the output terminals for the controller.
- **Input** (file 1) – This file stores the status of the input terminals for the controller.
- **Status** (file 2) – This file stores controller operation information. This file is useful for troubleshooting controller and program operation.
- **Bit** (file 3) – This file is used for internal relay logic storage.
- **Timer** (file 4) – This file stores the timer accumulator and preset values and status bits.

- **Counter** (file 5) – This file stores the counter accumulator and preset values and the status bits.
- **Control** (file 6) – This file stores the length, pointer position, and status bits for specific instructions such as shift registers and sequencers.
- **Integer** (file 7) – This file is used to store numeric values or bit information.

Understanding How Processor Files are Stored and Accessed

The MicroLogix 1000 programmable controller uses two devices for storing processor files: RAM and EEPROM. The RAM provides easy access storage (i.e., its data is lost on a power down), while the EEPROM provides long-term storage (i.e., its data is not lost on a power down). The diagram below shows how the memory is allocated in the micro controller's processor.

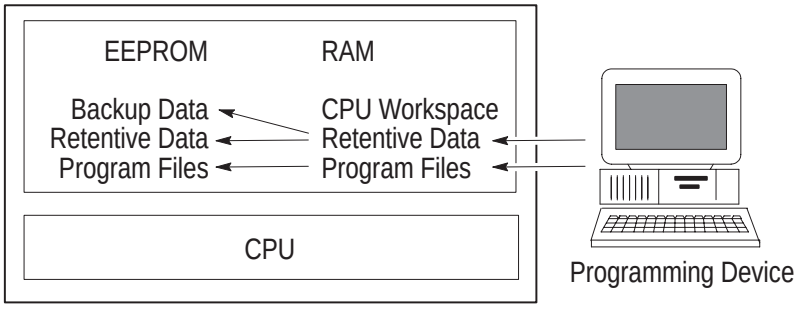


The memory device that is used depends on the operation being performed. This section describes how memory is stored and accessed during the following operations:

- download
- normal operation
- power down
- power up

Download

When the processor file is downloaded to the micro controller, it is first stored in the volatile RAM. It is then transferred to the non-volatile EEPROM, where it is stored as both backup data and retentive data.

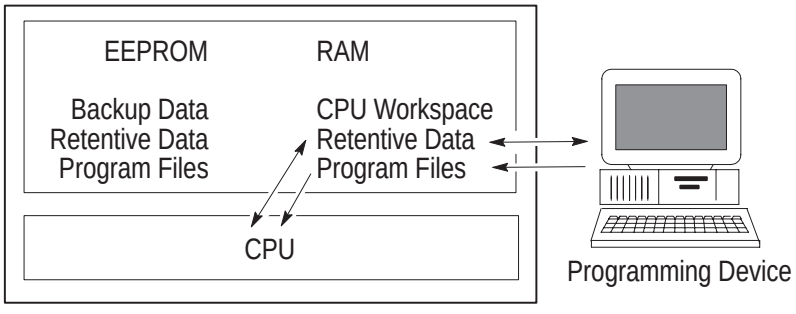


Note *If you want to ensure that the backup data is the same for every micro controller you are using, save the program to disk before downloading it to a micro controller.*

Normal Operation

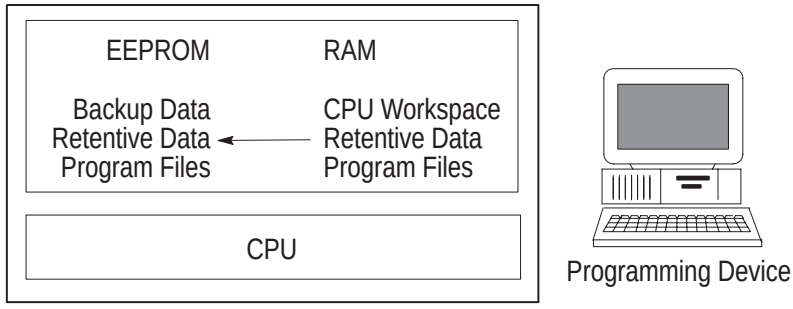
During normal operation, both the micro controller and your programming device can access the processor files stored in the RAM. Any changes to retentive data that occur due to program execution or programming commands affect only the retentive data in the RAM.

The program files are never modified during normal operation. However, both the CPU and your programming device can read the program files stored in RAM.



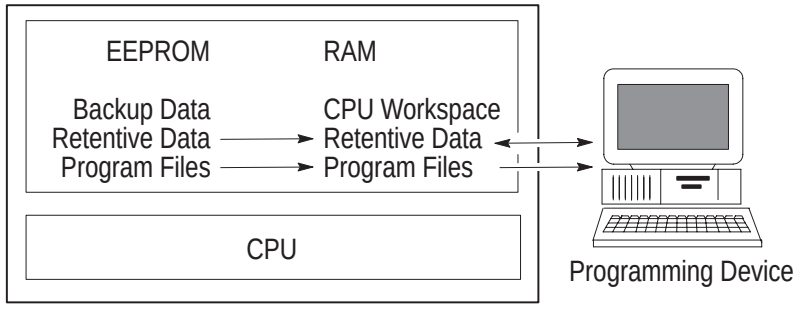
Power Down

When a power down occurs, only the retentive data is transferred from the RAM to the EEPROM. (The program files do not need to be saved to the EEPROM since they cannot be modified during normal operation.) If for some reason power is lost before all of the retentive data is saved to the EEPROM, the retentive data is lost. This may occur due to an unexpected reset or a hardware problem.

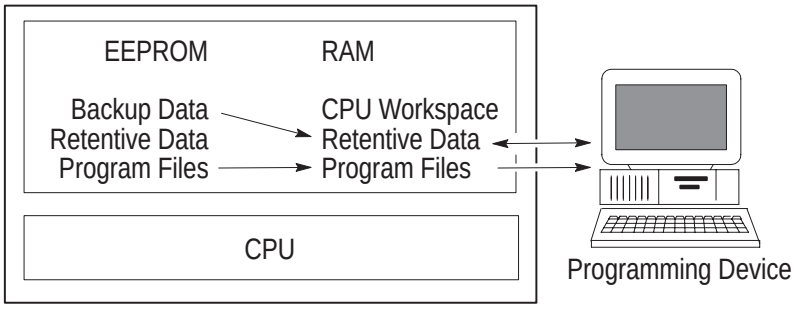


Power Up

During power up, the micro controller transfers the program files from the EEPROM to the RAM. The retentive data is also transferred to the RAM, provided it was not lost on power down, and normal operation begins.



If retentive data was lost on power down, the backup data from the EEPROM is transferred to the RAM and used as the retentive data. In addition, status file bit S2:5/8 (retentive data lost) is set and a recoverable major error occurs when going to run.



Addressing Data Files

For the purposes of addressing, each data file type is identified by a letter (identifier) and a file number.

File Type	Identifier	File Number
Output	O	0
Input	I	1
Status	S	2
Bit	B	3
Timer	T	4
Counter	C	5
Control	R	6
Integer	N	7

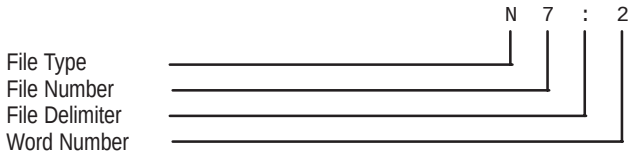
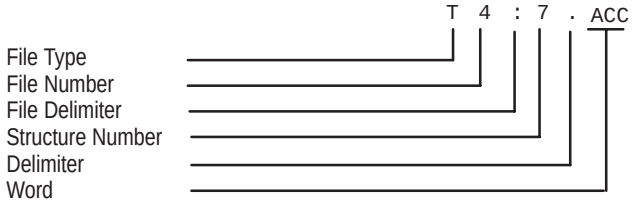
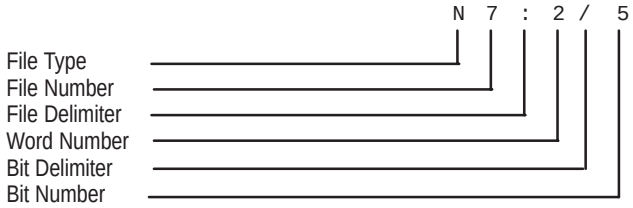
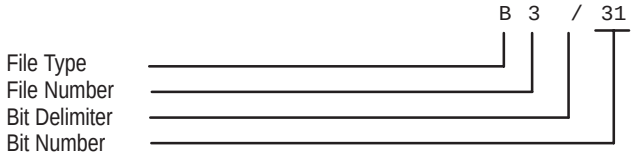
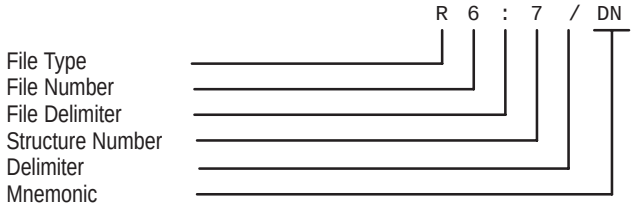
The addresses are made up of alphanumeric characters separated by delimiters. Delimiters include the colon, slash, and period.

Specifying Logical Addresses

The format of a logical address, **`xf:e`**, corresponds directly to the location in data storage.

Where:	Is the:
x	File type: O—output I—input S—status B—binary T—timer C—counter R—control N—integer
f	File #: 0—output 1—input 2—status 3—binary 4—timer 5—counter 6—control 7—integer
:	File delimiter: Colon or semicolon delimiter separates file and structure/word numbers.
e	Element number: 0 to: 0—output 1—input 32—status 31—binary 39—timer 31—counter 15—control 104—integer

You assign logical addresses to instructions from the highest level (element) to the lowest level (bit). Addressing examples are shown in the table below.

To specify the address of a:	Use these parameters:①
Word within an integer file	
Word within a structure file (e.g., a timer file)	
Bit within an integer file	
Bit within a bit file	 Bit files are bit stream continuous files, and therefore you can address them in two ways: by word and bit, or by bit alone.
Bit within a structure file (e.g., a control file)	

① Some programming devices support short addressing. This allows you to eliminate the file number and file delimiter from addresses. (For example: N7:2=N2, T4:12.ACC=T12.ACC, B3:2/12=B2/12) Consult your programming device's user manual for information on addressing capabilities.

You can also address at the bit level using mnemonics for timer, counter, or control data types. The available mnemonics depend on the type of data. See chapters 6 through 13 for more information.

Specifying Indexed Addresses

The indexed address symbol is the # character. Place the # character immediately before the file-type identifier in a logical address. You can use more than one indexed address in your ladder program.

Enter the offset value in word 24 of the status file (S:24). All indexed instructions use the same word S:24 to store the offset value. The processor starts operation at the base address plus the offset. You can manipulate the offset value in your ladder logic before each indexed address operation.

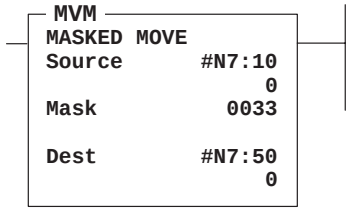
When you specify indexed addresses, follow these guidelines:

- Make sure the index value (positive or negative) does not cause the indexed address to exceed the file type boundary.
- When an instruction uses more than two indexed addresses, the processor uses the same index value for each indexed address.
- Set the index word to the offset value you want immediately before enabling an instruction that uses an indexed address.

Instructions with a # sign in an address manipulate the offset value stored at S:24. Make sure you monitor or load the offset value you want prior to using an indexed address. Otherwise unpredictable machine operation could occur with possible damage to equipment and/or injury to personnel.

Example of Indexed Addressing

The following Masked Move (MVM) example uses an indexed address in the source and destination addresses. If the offset value is 10 (stored in S:24), the processor manipulates the data stored at the base address plus the offset.



In this example, the processor uses the following addresses:

Value:	Base Address:	Offset Value in S:24	Offset Address:
Source	N7:10	10	N7:20
Destination	N7:50	10	N7:60

Addressing File Instructions – Using the File Indicator (#)

The file instructions below manipulate data table files. These files are addressed with the # sign. They store an offset value in word S:24 (index register), just as with indexed addressing discussed in the last section.

COP Copy File
FLL Fill File
BSL Bit Shift Left
BSR Bit Shift Right
FFL (FIFO Load)
FFU (FIFO Unload)
LFL (LIFO Load)
LFU (LIFO Unload)
SQO Sequencer Output
SQC Sequencer Compare
SQL Sequencer Load

If you are using file instructions and also indexed addressing, make sure that you monitor and/or load the correct offset value prior to using an indexed address. Otherwise, unpredictable operation could occur, resulting in possible personal injury and/or damage to equipment.

Numeric Constants

You can enter numeric constants directly into many of the instructions you program. The range of values for most instructions is –32,768 through +32,767. These values can be displayed or entered in several radixes. The radixes that can be *displayed* are:

- Integer
- Binary
- ASCII
- Hexadecimal

When *entering* values into an instruction or data table element, you can specify the radix of your entry using the "&" special operator. The radixes that can be used to enter data into an instruction or data table element are:

- Integer (&N)
- Binary (&B)
- ASCII (&A)
- Hexadecimal (&H)
- BCD (&D)
- Octal (&O)

Numeric constants are used in place of data file elements. They cannot be manipulated by the user program. You must enter the offline program editor to change the value of a constant.

Applying Ladder Logics to Your Schematics

The logic you enter into the micro controller makes up a ladder program. A ladder program consists of a set of instructions used to control a machine or a process.

Ladder logic is a graphical programming language based on electrical relay diagrams. Instead of having electrical rung continuity, ladder logic is looking for logical rung continuity. A ladder diagram identifies each of the elements in an electromechanical circuit and represents them graphically. This allows you to see how your control circuit operates before you actually start the physical operation of your system.

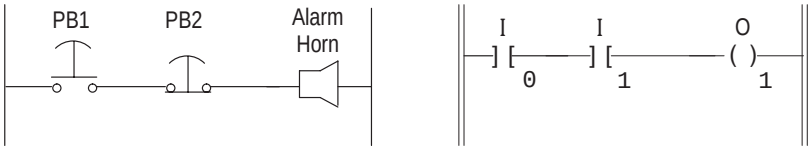
In a ladder diagram each of the input devices are represented in series or parallel combinations across the rung of the ladder. The last element on the rung is the output that receives the action as a result of the conditional state of the inputs on the rung.

Each output instruction is executed by the controller when the rung is scanned and the conditions on the rung are true. When the rung is not scanned or the logic conditions on the rung do not create a true logic path, the output is not executed.

The programming device allows you to enter a ladder logic program into the micro controller.

In the following illustration, the electromechanical circuit shows PB1 and PB2, two pushbuttons, wired in series with an alarm horn. PB1 is a normally open pushbutton, and PB2 is normally closed. This same circuit is shown in ladder logic by two contacts wired in series with an output. Contact I/0 and I/1 are examine-if-closed instructions.[1] (For more information on this instruction, refer to page 6–4.)

Electromechanical Circuit **Ladder Logic Program**



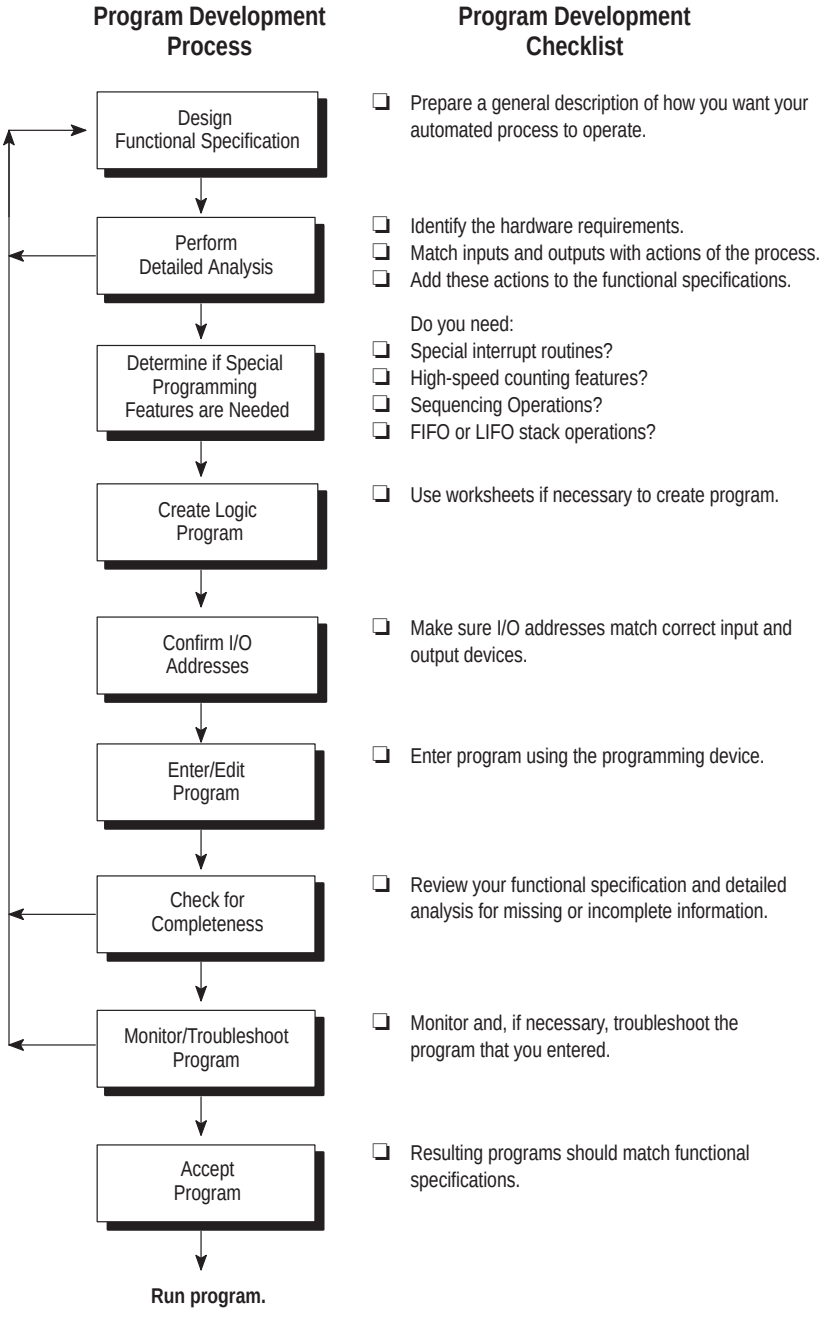
[1] Contact I1 would be an examine-if-open instruction ( ]/[ ) if PB2 was a normally open electromechanical circuit.

The table below shows how these circuits operate. The table shows all possible conditions for the electromechanical circuit, the equivalent state of the ladder logic instructions, and the resulting output state.

If PB1 is:	I/0 state is:	And PB2 is:	I/1 state is:	Then the Alarm Horn (O/1) is:
not pushed	0	not pushed	1	silent
not pushed	0	pushed	0	silent
pushed	1	not pushed	1	alarm
pushed	1	pushed	0	silent

Developing Your Logic Program – A Model

The following diagram can help you develop your application program. Each process block represents one phase of program development. Use the checklist at the right of the process block to help you identify the tasks involved with each process.


Program Development Process
Program Development Checklist
Design Functional Specification
Perform Detailed Analysis
Determine if Special Programming Features are Needed
Create Logic Program
Confirm I/O Addresses
Enter/Edit Program
Check for Completeness
Monitor/Troubleshoot Program
Accept Program
Run program.
❑ Prepare a general description of how you want your automated process to operate.
❑ Identify the hardware requirements.
❑ Match inputs and outputs with actions of the process.
❑ Add these actions to the functional specifications.
Do you need:
❑ Special interrupt routines?
❑ High-speed counting features?
❑ Sequencing Operations?
❑ FIFO or LIFO stack operations?
❑ Use worksheets if necessary to create program.
❑ Make sure I/O addresses match correct input and output devices.
❑ Enter program using the programming device.
❑ Review your functional specification and detailed analysis for missing or incomplete information.
❑ Monitor and, if necessary, troubleshoot the program that you entered.
❑ Resulting programs should match functional specifications.

5 Using Analog

This chapter describes the operation of the MicroLogix 1000 analog controllers. Topics include:

- I/O Image
- I/O Configuration
- Input Filter and Update Times
- Converting Analog Data

I/O Image

The input and output image files of the MicroLogix 1000 analog controllers have the following format:

Address	Input Image	Output Image	Address
I:0.0	Discrete Input Word 0	Discrete Output Word 0	O:0.0
I:0.1	Discrete Input Word 1	Reserved	O:0.1
I:0.2	Reserved	Reserved	O:0.2
I:0.3	Reserved	Reserved	O:0.3
I:0.4	Analog Input 0 (Voltage)	Analog Output 0 (Voltage or Current)	O:0.4
I:0.5	Analog Input 1 (Voltage)		
I:0.6	Analog Input 2 (Current)		
I:0.7	Analog Input 3 (Current)		

Input words 0 and 1 contain discrete input data. Unused inputs in the discrete inputs image space are reset during each input scan. Input words 2 and 3 are reserved and are not updated by the controller. These inputs have no direct effect on controller operation, but they can be modified like other data bits.

Input words 4–7 contain the status of the four analog input channels respectively. Analog input image words are cleared at Going To Run (GTR). For enabled channels, the analog input image is updated on a cyclical basis.

Output word 0 contains discrete output data. Output words 1–3 are reserved output image space. Unused outputs in both the discrete output image space and the reserved output image space have no direct effect on controller operation. But these outputs can be modified like other data bits. Output word 4 holds the value of the analog output channel.

I/O Configuration

The analog input channels are single-ended (unipolar) circuits and can be individually enabled or disabled. The default is all input channels enabled. The two voltage inputs accept ± 10.5V dc, and the two current inputs accept ± 21 mA.

The analog output channel is also a single-ended circuit. You can configure either voltage (0V dc to +10V dc) or current (+4 to +20 mA) output operation. The default is voltage output.

The output must be configured for either voltage *or* current, *not both*. This is determined by the output configuration. When in the Run mode and the output is configured for voltage, the voltage output terminal is active and the current output terminal is inactive. Similarly, when in the Run mode and the output is configured for current, the current output terminal is active and the voltage output terminal is inactive. When the system is not in Run mode, both the voltage and current outputs are inactive.

Input Filter and Update Times

The MicroLogix analog input filter is programmable. The slower the filter setting, the more immune the analog inputs are to electrical noise. The more immune the analog inputs are to electrical noise, the slower the inputs will be to update. Similarly, the faster the filter setting, the less immune the analog inputs are to electrical noise. The less immune the analog inputs are to electrical noise, the faster the inputs will be to update.

Programmable Filter Characteristics				
1st Notch Freq (Hz)	Filter Bandwidth (-3 dB Freq Hz)	Update Time (mSec)	Settling Time (mSec)	Resolution (Bits)
10	2.62	100.00	400.00	16
50	13.10	20.00	80.00	16
60①	15.72	16.67	66.67	16
250	65.50	4.00	16.00	15

① 60 Hz is the default setting.

The total update time for each channel is a combination of the Update Time and the Settling Time. When more than one analog input channel is enabled, the maximum update for each channel is equal to one ladder scan time plus the channel's Update Time plus Settling Time. When only one analog input channel is enabled, the maximum update for the channel is equal to the Update Time plus one ladder scan time.

Update Examples

Example 1 – All (4) channels enabled with 60 Hz filter selected (default settings).

Maximum Update Time = (4) x ladder scan time
+ (4) x 16.67 ms
+ (4) x 66.67 ms
= 333.36 ms + (4) x ladder scan times

(Each channel will be updated approximately three times per second.)

Example 2 – 1 channel enabled with 250 Hz filter selected.

Maximum Update Time = ladder scan time + 4 ms

Input Channel Filtering

The analog input channels incorporate on-board signal conditioning. The purpose of this conditioning is to reject the AC power line noise that can couple into an analog input signal while passing the normal variations of the input signal.

Frequency components of the input signal at the filter frequency are rejected. Frequency components below the filter bandwidth (–3 dB frequency) are passed with under 3 dB of attenuation. This pass band allows the normal variation of sensor inputs such as temperature, pressure and flow transducers to be input data to the processor.

Noise signals coupled in at frequencies above the pass band are sharply rejected. An area of particular concern is the 50/60 Hz region, where pick-up from power lines can occur.

Converting Analog Data

The analog input circuits are able to monitor current and voltage signals and convert them to digital data. There are six terminals assigned to the input channels that provide two voltage inputs, two current inputs and two return signals (commons).

The analog outputs can support either a current *or* voltage function. There are three terminals assigned to the output channels that provide one voltage output, one current output and a common (shared) terminal.

The following table shows sample Analog Signal and Data Word values using the nominal transfer function formula:

$N=I_{in} \times 32767/21$ where I_{in} (analog signal) is in milliamperes (mA)

$N=V_{in} \times 32767/10.5$ where Vin (analog signal) is in volts (V)

$N=(I_{out} - 4\ mA) \times 32767/16\ mA$ where I_{out} (analog signal) is in milliamperes (mA)

$N=V_{out} \times 32767/10V$ where V_{out} (analog signal) is in volts (V)

Analog Signal	Data Word	
	Input	Output
0V	0	0
5V	15603	16384
10V	31207	32767
4 mA	6241	0
11 mA	17164	14336
20 mA	31207	32767

Converting Analog Input Data

Analog inputs convert current and voltage signals into 16-bit two's complement binary values. To determine an approximate voltage that an input value represents, use one of the equations shown on the following page.

$$\frac{10.5V}{32{,}767} \times \text{input value}^{①} = \text{input voltage(V)}$$

①The Input Value is the decimal value of the word in the input image for the corresponding analog input.

For example, if an input value of 16,021 is in the input image, the calculated value is:

$$\frac{10.5V}{32{,}767} \times 16{,}201 = 5.1915(V)$$

It should be noted that the actual value may vary within the accuracy limitations of the module.

To determine an approximate current that an input value represents, you can use the following equation:

$$\frac{21\ mA}{32{,}767} \times \text{input value}^{②} = \text{input current (mA)}$$

②The Input Value is the decimal value of the word in the input image for the corresponding analog input.

For example, if an input value of 4096 is in the input image, the calculated value is:

$$\frac{21\ mA}{32{,}767} \times 4096 = 2.625\ (mA)$$

It should be noted that the actual value may vary within the accuracy limitations of the module.

Converting Analog Output Data

Use the following equation to determine the decimal value for the current output:

$$\frac{32{,}767}{16\ mA} \times [\text{Desired Current Output (mA)} - 4\ mA] = \text{Output Decimal Value}$$

For example, if an output value of 8 mA is desired, the value to be put in the corresponding word in the output image can be calculated as follows:

$$\frac{32{,}767}{16\ mA} \times (8\ mA - 4\ mA) = 8192$$

Use the following equation to determine the decimal value for the voltage output:

$$\frac{32{,}767}{10V\ dc} \times \text{Desired Voltage Output (V dc)} = \text{Output Decimal Value}$$

For example, if an output value of 1V dc is desired, the value to be put in the corresponding word in the output image can be calculated as follows:

$$\frac{32{,}767}{10V\ dc} \times 1V\ dc = 3277$$

6 Using Basic Instructions

This chapter contains general information about the basic instructions and explains how they function in your application program. Each of the basic instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the basic instructions in use.

Bit Instructions

Instruction		Purpose	Page
Mnemonic	Name		
XIC	Examine if Closed	Examines a bit for an On condition.	**6–4**
XIO	Examine if Open	Examines a bit for an Off condition.	**6–4**
OTE	Output Energize	Turns a bit On or Off.	**6–5**
OTL and OTU	Output Latch and Output Unlatch	OTL turns a bit on when the rung is executed, and this bit retains its state when the rung is not executed or a power cycle occurs. OTU turns a bit off when the rung is executed, and this bit retains its state when the rung is not executed or when power cycle occurs.	**6–5**
OSR	One-Shot Rising	Triggers a one time event.	**6–7**

Timer/Counter Instructions

Instruction		Purpose	Page
Mnemonic	Name		
TON	Timer On-Delay	Counts timebase intervals when the instruction is true.	**6–11**
TOF	Timer Off-Delay	Counts timebase intervals when the instruction is false.	**6–12**
RTO	Retentive Timer	Counts timebase intervals when the instruction is true and retains the accumulated value when the instruction goes false or when power cycle occurs.	**6–14**
CTU	Count Up	Increments the accumulated value at each false-to-true transition and retains the accumulated value when the instruction goes false or when power cycle occurs.	**6–18**
CTD	Count Down	Decrements the accumulate value at each false-to-true transition and retains the accumulated value when the instruction goes false or when power cycle occurs.	**6–19**
RES	Reset	Resets the accumulated value and status bits of a timer or counter. Do not use with TOF timers.	**6–20**

About the Basic Instructions

These instructions, when used in ladder programs represent hardwired logic circuits used for the control of a machine or equipment.

The basic instructions are separated into three groups: bit, timer, and counter. Before you learn about the instructions in each of these groups, we suggest that you read the overview that precedes the group:

- Bit Instructions Overview
- Timer Instructions Overview
- Counter Instructions Overview

Bit Instructions Overview

These instructions operate on a single bit of data. During operation, the controller may set or reset the bit, based on the logical continuity of ladder rungs. You can address a bit as many times as your program requires.

Note *Using the same address with multiple output instructions is not recommended.*

Bit instructions are used with the following data files:

- Output and input data files. These represent external outputs and inputs.
- The status data file (file 2).
- The bit data file (B3:). These are the internal coils used in your program.
- Timer, counter, and control data files (T4:, C5:, and R6:). These instructions use various control bits.
- The integer data file (N7:). Use these addresses (at the bit level) as your program requires.

Examine if Closed (XIC)

—] [—

Execution Times (µsec) when:

True	False
1.54	1.72

Use the XIC instruction in your ladder program to determine if a bit is On. When the instruction is executed, if the bit addressed is on (1), then the instruction is evaluated as true. When the instruction is executed, if the bit addressed is off (0), then the instruction is evaluated as false.

Bit Address State	XIC Instruction
0	False
1	True

Examples of devices that turn on or off include:

- a push button wired to an input (addressed as I1:0/4)
- an output wired to a pilot light (addressed as O0:0/2)
- a timer controlling a light (addressed as T4:3/DN)

Examine if Open (XIO)

—]/[—

Execution Times (µsec) when:

True	False
1.54	1.72

Use an XIO instruction in your ladder program to determine if a bit is Off. When the instruction is executed, if the bit addressed is off (0), then the instruction is evaluated as true. When the instruction is executed, if the bit addressed is on (1), then the instruction is evaluated as false.

Bit Address State	XIO Instruction
0	True
1	False

Examples of devices that turn on or off include:

- motor overload normally closed (N.C.) wired to an input (I1:0/10)
- an output wired to a pilot light (addressed as O0:0/4)
- a timer controlling a light (addressed as T4:3/DN)

Output Energize (OTE)

—( )—

Use an OTE instruction in your ladder program to turn On a bit when rung conditions are evaluated as true.

Execution Times (µsec) when:

True	False
4.43	4.43

An example of a device that turns on or off is an output wired to a pilot light (addressed as O0:0/4).

OTE instructions are reset when:

- You enter or return to the REM Run or REM Test mode or power is restored.
- The OTE is programmed within an inactive or false Master Control Reset (MCR) zone.

Note

A bit that is set within a subroutine using an OTE instruction remains set until the subroutine is scanned again.

Output Latch (OTL) and Output Unlatch (OTU)

—(L)—

—(U)—

OTL and OTU are retentive output instructions. OTL can only turn on a bit, while OTU can only turn off a bit. These instructions are usually used in pairs, with both instructions addressing the same bit.

Your program can examine a bit controlled by OTL and OTU instructions as often as necessary.

Execution Times (µsec) when:

	True	False
OTL	4.97	3.16
OTU	4.97	3.16

Under fatal error conditions, physical outputs are turned off. Once the error conditions are cleared, the controller resumes operation using the data table value of the operand.

Using OTL

When you assign an address to the OTL instruction that corresponds to the address of a physical output, the output device wired to this screw terminal is energized when the bit is set (turned on or enabled).

When rung conditions become false (after being true), the bit remains set and the corresponding output device remains energized.

When enabled, the latch instruction tells the controller to turn on the addressed bit. Thereafter, the bit remains on, regardless of the rung condition, until the bit is turned off (typically by a OTU instruction in another rung).

Using OTU

When you assign an address to the OTU instruction that corresponds to the address of a physical output, the output device wired to this screw terminal is de-energized when the bit is cleared (turned off or disabled).

The unlatch instruction tells the controller to turn off the addressed bit. Thereafter, the bit remains off, regardless of the rung condition, until it is turned on (typically by a OTL instruction in another rung).

One-Shot Rising (OSR)

```
──[OSR]──
```

Execution Times (µsec) when:

True	False
13.02	11.48

The OSR instruction is a retentive input instruction that triggers an event to occur one time. Use the OSR instruction when an event must start based on the change of state of the rung from false to true.

When the rung conditions preceding the OSR instruction go from false to true, the OSR instruction will be true for one scan. After one scan is complete, the OSR instruction becomes false, even if the rung conditions preceding it remain true. The OSR instruction will only become true again if the rung conditions preceding it transition from false to true.

The controller allows you to use one OSR instruction per output in a rung.

Entering Parameters

The address assigned to the OSR instruction is *not* the one-shot address referenced by your program, nor does it indicate the state of the OSR instruction. This address allows the OSR instruction to *remember* its previous rung state.

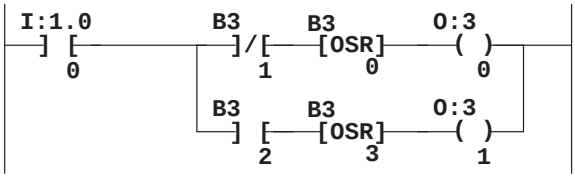
Use a bit address from either the bit or integer data file. The addressed bit is set (1) for one scan when rung conditions preceding the OSR instruction are true (even if the OSR instruction becomes false); the bit is reset (0) when rung conditions preceding the OSR instruction are false.

Note

The bit address you use for this instruction must be unique. Do not *use it elsewhere in the program.*

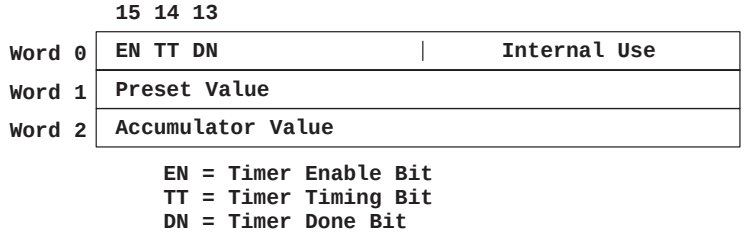
Do not *use an input or output address to program the address parameter of the OSR instruction.*

Example Rung



Timer Instructions Overview

Each timer address is made of a 3-word element. Word 0 is the control word, word 1 stores the preset value, and word 2 stores the accumulated value.



Entering Parameters

Accumulator Value (ACC)

This is the time elapsed since the timer was last reset. When enabled, the timer updates this continually.

Preset Value (PRE)

Specifies the value which the timer must reach before the controller sets the done bit. When the accumulated value becomes equal to or greater than the preset value, the done bit is set. You can use this bit to control an output device.

Preset and accumulated values for timers range from 0 to +32,767. If a timer preset or accumulated value is a negative number, a runtime error occurs.

Timebase

The timebase determines the duration of each timebase interval. The timebase is selectable as 0.01 (10 ms) second or 1.0 second.

Timer Accuracy

Timer accuracy refers to the length of time between the moment a timer instruction is enabled and the moment the timed interval is complete.

Timing accuracy is –0.01 to +0 seconds, with a program scan of up to 2.5 seconds. The 1-second timer maintains accuracy with a program scan of up to 1.5 seconds. If your programs can exceed 1.5 or 2.5 seconds, repeat the timer instruction rung so that the rung is scanned within these limits.

Note *Timing could be inaccurate if Jump (JMP), Label (LBL), Jump to Subroutine (JSR), or Subroutine (SBR) instructions skip over the rung containing a timer instruction while the timer is timing. If the skip duration is within 2.5 seconds, no time will be lost; if the skip duration exceeds 2.5 seconds, an undetectable timing error occurs. When using subroutines, a timer must be executed at least every 2.5 seconds to prevent a timing error.*

Addressing Structure

Address bits and words using the format **Tf:e.s/b**

Format	Explanation		
Tf:e	**T**	Timer file	
	f	File number. The only valid file number is 4.	
	:	Element delimiter	
	e	Element number	Ranges from 0 – 39. These are 3-word elements. See figure on page 6–8.
	.	Word element	
	s	subelement	
	/	Delimiter	
	b	bit	

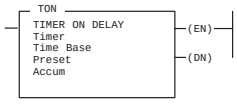
Addressing Examples

- **T4:0/15** or **T4:0/EN** Enable bit
- **T4:0/14** or **T4:0/TT** Timer timing bit
- **T4:0/13** or **T4:0/DN** Done bit

- **T4:0.1** or **T4:0.PRE** Preset value of the timer
- **T4:0.2** or **T4:0.ACC** Accumulator value of the timer

- **T4:0.1/0 or T4:0.PRE/0** Bit 0 of the preset value
- **T4:0.2/0 or T4:0.ACC/0** Bit 0 of the accumulated value

Timer On-Delay (TON)



Execution Times (µsec) when:

True	False
38.34	30.38

Use the TON instruction to delay the turning on or off of an output. The TON instruction begins to count timebase intervals when rung conditions become true. As long as rung conditions remain true, the timer increments its accumulated value (ACC) each scan until it reaches the preset value (PRE). The accumulated value is reset when rung conditions go false, regardless of whether the timer has timed out.

Using Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Timer Done Bit DN (bit 13)	accumulated value is equal to or greater than the preset value	rung conditions go false
Timer Enable Bit EN (bit 14)	rung conditions are true	rung conditions go false
Timer Timing Bit TT (bit 15)	rung conditions are true and the accumulated value is less than the preset value	rung conditions go false or when the done bit is set

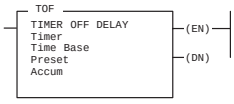
When the controller changes from the REM Run or REM Test mode to the REM Program mode or user power is lost while the instruction is timing but has not reached its preset value, the following occurs:

- Timer Enable (EN) bit remains set.
- Timer Timing (TT) bit remains set.
- Accumulated value (ACC) remains the same.

On returning to the REM Run or REM Test mode, the following can happen:

Condition	Result
If the rung is true:	EN bit remains set. TT bit remains set. ACC value is reset.
If the rung is false:	EN bit is reset. TT bit is reset. ACC value is reset.

Timer Off-Delay (TOF)



Execution Times (µsec) when:

True	False
39.42	31.65

Use the TOF instruction to delay turning on or off an output. The TOF instruction begins to count timebase intervals when the rung makes a true-to-false transition. As long as rung conditions remain false, the timer increments its accumulated value (ACC) each scan until it reaches the preset value (PRE). The controller resets the accumulated value when rung conditions go true regardless of whether the timer has timed out.

Using Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Timer Done Bit DN (bit 13)	rung conditions are true	rung conditions go false and the accumulated value is greater than or equal to the preset value
Timer Timing Bit TT (bit 14)	rung conditions are false and the accumulated value is less than the preset value	rung conditions go true or when the done bit is reset
Timer Enable Bit EN (bit 15)	rung conditions are true	rung conditions go false

When the controller changes from the REM Run or REM Test mode to the REM Program mode, or user power is lost while a timer off-delay instruction is timing but has not reached its preset value, the following occurs:

- Timer Enable (EN) bit remains set.
- Timer Timing (TT) bit remains set.
- Timer Done (DN) bit remains set.
- Accumulated value (ACC) remains the same.

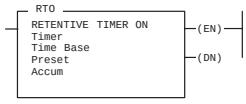
On returning to the REM Run or REM Test mode, the following can happen:

Condition	Result
If the rung is true:	TT bit is reset. DN bit remains set. EN bit is set. ACC value is reset.
If the rung is false:	TT bit is reset. DN bit is reset. EN bit is reset. ACC value is set equal to the preset value.

The Reset (RES) instruction cannot be used with the TOF instruction because RES always clears the status bits as well as the accumulated value. (See page 6–20.)

Note *The TOF times inside an inactive MCR Pair.*

Retentive Timer (RTO)



Use the RTO instruction to turn an output on or off after its timer has been on for a preset time interval. The RTO instruction is a retentive instruction that lets the timer stop and start without resetting the accumulated value (ACC).

Execution Times (µsec) when:

True	False
38.34	27.49

The RTO instruction retains its accumulated value when any of the following occurs:

- Rung conditions become false.
- You change controller operation from the REM Run or REM Test mode to the REM Program mode.
- The controller loses power.
- A fault occurs.

Using Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Timer Done Bit DN (bit 13)	accumulated value is equal to or greater than the preset value	the appropriate RES instruction is enabled
Timer Timing Bit TT (bit 14)	rung conditions are true and the accumulated value is less than the preset value	rung conditions go false or when the done bit is set
Timer Enable Bit EN (bit 15)	rung conditions are true	rung conditions go false

Note

To reset the retentive timer's accumulated value and status bits after the RTO rung goes false, you must program a reset (RES) instruction with the same address in another rung.

When the controller changes from the REM Run or REM Test mode to the REM Program or REM Fault mode, or user power is lost while the timer is timing but not yet at the preset value, the following occurs:

- Timer Enable (EN) bit remains set.
- Timer Timing (TT) bit remains set.
- Accumulated value (ACC) remains the same.

On returning to the REM Run or REM Test mode or when power is restored, the following can happen:

Condition	Result
If the rung is true:	TT bit remains set. EN bit remains set. ACC value remains the same and resumes incrementing.
If the rung is false:	TT bit is reset. DN bit remains in its last state. EN bit is reset. ACC value remains in its last state.

Counter Instructions Overview

Each Counter address is made of a 3-word data file element. Word 0 is the control word, containing the status bits of the instruction. Word 1 is the preset value. Word 2 is the accumulated value.

The control word for counter instructions includes six status bits, as indicated below.

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Word 0	CU	CD	DN	OV	UN	UA			Not Used							
Word 1	Preset Value															
Word 2	Accumulator Value															

CU = Counter up enable bit
CD = Counter down enable bit
DN = Done bit
OV = Overflow bit
UN = Underflow bit
UA = Update accumulator (HSC only)

For high-speed counter instruction information, see chapter 12.

Entering Parameters

Accumulator Value (ACC)

This is the number of false-to-true transitions that have occurred since the counter was last reset.

Preset Value (PRE)

Specifies the value which the counter must reach before the controller sets the done bit. When the accumulator value becomes equal to or greater than the preset value, the done status bit is set. You can use this bit to control an output device.

Preset and accumulated values for counters range from –32,768 to +32,767, and are stored as signed integers. Negative values are stored in two's complement form.

Addressing Structure

Address bits and words using the format **Cf:e.s/b**

Format	Explanation		
Cf:e	**C**	Counter file	
	f	File number. The only valid file number is 5.	
	:	Element delimiter	
	e	Element number	Ranges from 0 – 39. These are 3-word elements. See figure on page 6–15.
	.	Word element	
	s	subelement	
	/	Delimiter	
	b	bit	

Note *If assigned to a high-speed counter instruction, C5:0 is not available as an address for any other counter instructions. For more information on high-speed counter instructions, see chapter 12.*

Addressing Examples

- **C5:0/15** or **C5:0/CU** Count up enable bit
- **C5:0/14** or **C5:0/CD** Count down enable bit
- **C5:0/13** or **C5:0/DN** Done bit
- **C5:0/12** or **C5:0/OV** Overflow bit
- **C5:0/11** or **C5:0/UN** Underflow bit
- **C5:0/10** or **C5:0/UA** Update accumulator bit

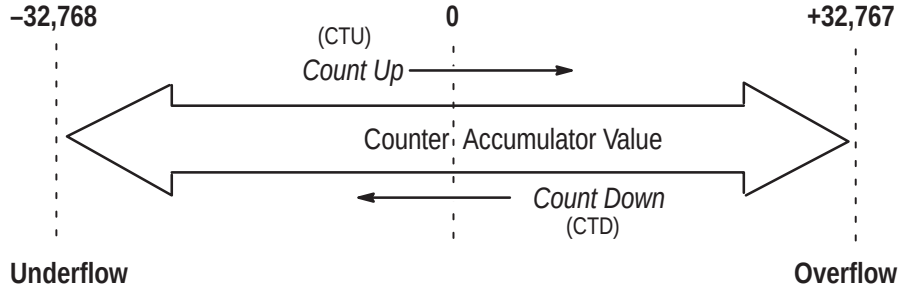
- **C5:0.1** or **C5:0.PRE** Preset value of the counter
- **C5:0.2** or **C5:0.ACC** Accumulator value of the counter

- **C5:0.1/0 or C5:0.PRE/0** Bit 0 of the preset value
- **C5:0.2/0 or C5:0.ACC/0** Bit 0 of the accumulated value

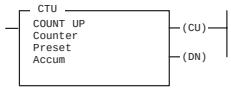
How Counters Work

The figure below demonstrates how a counter works. The count value must remain in the range of –32,768 to +32,767. If the count value goes above +32,767 or below –32,768, a counter status overflow (OV) or underflow (UN) bit is set.

A counter can be reset to zero using the reset (RES) instruction. (See page 6–20.)



Count Up (CTU)



Execution Times (µsec) when:

True	False
29.84	26.67

The CTU is an instruction that counts false-to-true rung transitions. Rung transitions can be caused by events occurring in the program (from internal logic or by external field devices) such as parts traveling past a detector or actuating a limit switch.

When rung conditions for a CTU instruction have made a false-to-true transition, the accumulated value is incremented by one count, provided that the rung containing the CTU instruction is evaluated between these transitions. The ability of the counter to detect false-to-true transitions depends on the speed (frequency) of the incoming signal.

Note *The on and off duration of an incoming signal must not be faster than the scan time 2x (assuming a 50% duty cycle).*

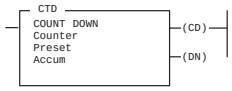
The accumulated value is retained when the rung conditions again become false. The accumulated count is retained until cleared by a reset (RES) instruction that has the same address as the counter reset.

Using Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Count Up Overflow Bit OV (bit 12)	accumulated value wraps around to –32,768 (from +32,767) and continues counting up from there	a RES instruction having the same address as the CTU instruction is executed OR the count is decremented less than or equal to +32,767 with a CTD instruction
Done Bit DN (bit 13)	accumulated value is equal to or greater than the preset value	the accumulated value becomes less than the preset
Count Up Enable Bit CU (bit 15)	rung conditions are true	rung conditions go false OR a RES instruction having the same address as the CTU instruction is enabled

The accumulated value is retained after the CTU instruction goes false, or when power is removed from and then restored to the controller. Also, the on or off status of counter done, overflow, and underflow bits is retentive. The accumulated value and control bits are reset when the appropriate RES instruction is enabled. The CU bits are always set prior to entering the REM Run or REM Test modes.

Count Down (CTD)



The CTD is a retentive output instruction that counts false-to-true rung transitions. Rung transitions can be caused by events occurring in the program such as parts traveling past a detector or actuating a limit switch.

Execution Times (µsec) when:

True	False
32.19	27.22

When rung conditions for a CTD instruction have made a false-to-true transition, the accumulated value is decremented by one count, provided that the rung containing the CTD instruction is evaluated between these transitions.

The accumulated counts are retained when the rung conditions again become false. The accumulated count is retained until cleared by a reset (RES) instruction that has the same address as the counter reset.

Using Status Bits

This Bit	Is Set When	And Remains Set Until One of the Following
Count Down Underflow Bit UN (bit 11)	accumulated value wraps around to +32,768 (from –32,767) and continues counting down from there	a RES instruction having the same address as the CTD instruction is enabled. OR the count is incremented greater than or equal to +32,767 with a CTU instruction
Done Bit DN (bit 13)	accumulated value is equal to or greater than the preset value	the accumulated value becomes less than the preset
Count Down Enable Bit CD (bit 14)	rung conditions are true	rung conditions go false OR a RES instruction having the same address as the CTD instruction is enabled

The accumulated value is retained after the CTD instruction goes false, or when power is removed from and then restored to the controller. Also, the on or off status of counter done, overflow, and underflow bits is retentive. The accumulated value and control bits are reset when the appropriate RES instruction is executed. The CD bits are always set prior to entering the REM Run or REM Test modes.

Reset (RES)

—(RES)—

Execution Times (µsec) when:

True	False
15.19	4.25

Use a RES instruction to reset a timer or counter. When the RES instruction is executed, it resets the data having the same address as the RES instruction.

Using a RES instruction for a:	The controller resets the:
Timer (Do not use a RES instruction with a TOF.)	ACC value to 0 DN bit TT bit EN bit
Counter	ACC value to 0 OV bit UN bit DN bit CU bit CD bit
Control	POS value to 0 EN bit EU bit DN bit EM bit ER bit UL bit IN and FD go to last state

Note *If using this instruction to reset the HSC accumulator, see page 12–21.*

When resetting a counter, if the RES instruction is enabled and the counter rung is enabled, the CU or CD bit is reset.

If the counter preset value is negative, the RES instruction sets the accumulated value to zero. This in turn causes the done bit to be set by a count down or count up instruction.

Because the RES instruction resets the accumulated value, and the done, timing, and enabled bits, do *not* use the RES instruction to reset a timer address used in a TOF instruction. Otherwise, unpredictable machine operation or injury to personnel may occur.

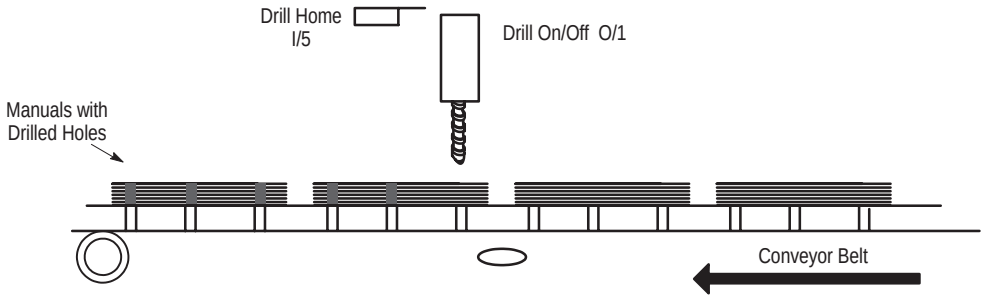
Basic Instructions in the Paper Drilling Machine Application Example

This section provides ladder rungs to demonstrate the use of basic instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will be adding the main program in file 2 and adding a subroutine to file 6.

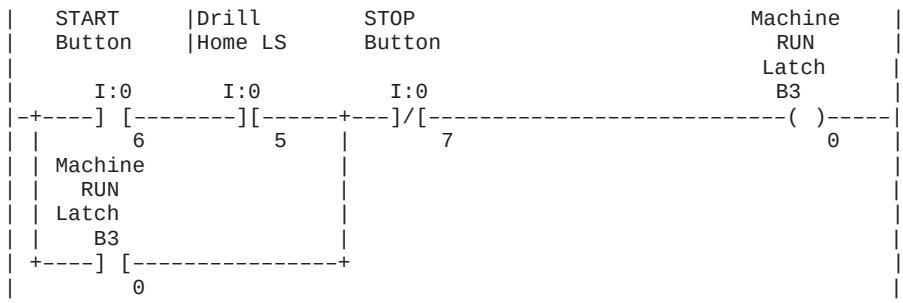
Adding File 2

The rungs shown on the following page are referred to as the program's "start-up" logic. They determine the conditions necessary to start the machine in motion by monitoring the start and stop push buttons. When the start push button is pressed, it enables the conveyor to move and starts spinning the drill bit. When the stop push button is pressed, it disables the conveyor motion and turns off the drill motor.

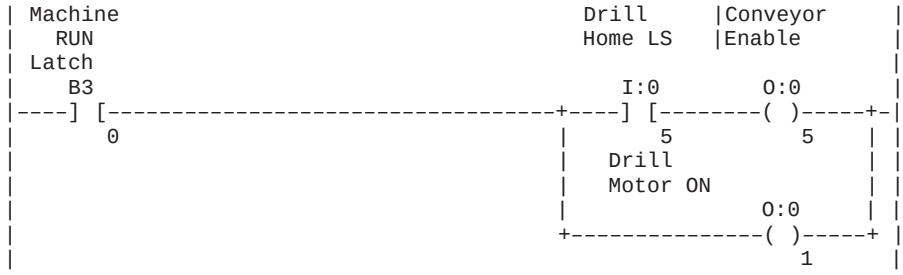
The start-up logic also checks to make sure that the drill is fully retracted (in the home position) before allowing the conveyor to move.



Rung 2:3[1]
Starts the conveyor in motion when the start button is pressed. However, another condition must also be met before we start the conveyor: the drill must be in its fully retracted position (home). This rung also stops the conveyor when the stop button is pressed.



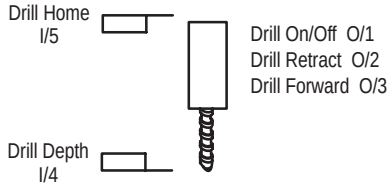
Rung 2:4
Applies the above start logic to the conveyor and drill motor.



[1] Rungs 2:0 through 2:2 will be added in chapter 12.

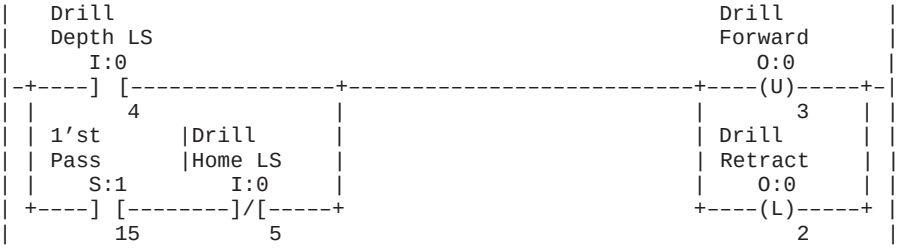
Adding File 6

This subroutine controls the up and down motion of the drill for the paper drilling machine.

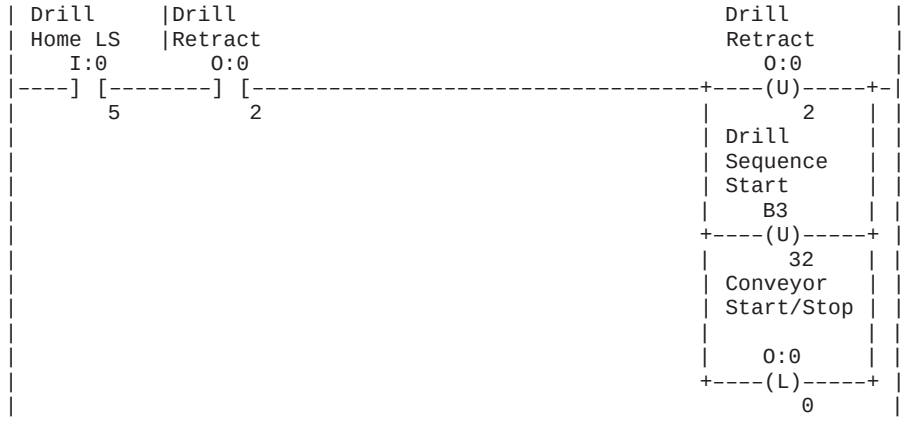



```
Rung 6:0
This section of ladder logic controls the up/down motion of the drill
for the book drilling machine.  When the conveyor positions the book
under the drill, the DRILL SEQUENCE START bit is set.  This rung uses
that bit to begin the drilling operation.  Because the bit is set for
the entire drilling operation, the OSR is required to be able to turn
off the forward signal so the drill can retract.
```

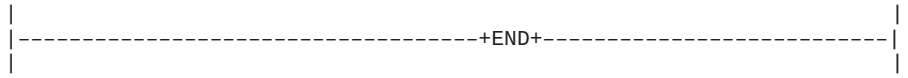
```
Rung 6:1
When the drill has drilled through the book, the body of the drill
actuates the DRILL DEPTH limit switch.  When this happens, the DRILL
FORWARD signal is turned off and the DRILL RETRACT signal is turned
on.  The drill is also retracted automatically on power up if it is
not actuating the DRILL HOME limit switch.
```



Rung 6:2
When the drill is retracting (after drilling a hole), the body of the drill actuates the DRILL HOME limit switch. When this happens the DRILL RETRACT signal is turned off, the DRILL SEQUENCE START bit is turned off to indicate the drilling process is complete, and the conveyor is restarted.



Rung 6.3



7 Using Comparison Instructions

This chapter contains general information about comparison instructions and explains how they function in your application program. Each of the comparison instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the comparison instructions in use.

Comparison Instructions

Instruction		Purpose	Page
Mnemonic	Name		
EQU	Equal	Test whether two values are equal.	**7–3**
NEQ	Not Equal	Test whether one value is not equal to a second value.	**7–3**
LES	Less Than	Test whether one value is less than a second value.	**7–3**
LEQ	Less Than or Equal	Test whether one value is less than or equal to a second value.	**7–4**
GRT	Greater Than	Test whether one value is greater than another.	**7–4**
GEQ	Greater Than or Equal	Test whether one value is greater than or equal to a second value.	**7–4**
MEQ	Masked Comparison for Equal	Test portions of two values to see whether they are equal. Compares 16-bit data of a source address to 16-bit data at a reference address through a mask.	**7–5**
LIM	Limit Test	Test whether one value is within the limit range of two other values.	**7–6**

About the Comparison Instructions

Comparison instructions are used to test pairs of values to condition the logical continuity of a rung. As an example, suppose a LES instruction is presented with two values. If the first value is less than the second, then the comparison instruction is true.

To learn more about the compare instructions, we suggest that you read the Compare Instructions Overview that follows.

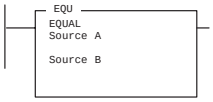
Comparison Instructions Overview

The following general information applies to comparison instructions.

Indexed Word Addresses

When using comparison instructions, you have the option of using indexed word addresses for instruction parameters specifying word addresses. Indexed addressing is discussed in chapter 5.

Equal (EQU)



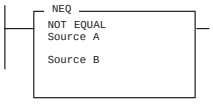
Use the EQU instruction to test whether two values are equal. If source A and source B are equal, the instruction is logically true. If these values are not equal, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Execution Times (µsec) when:

True	False
21.52	6.60

Not Equal (NEQ)



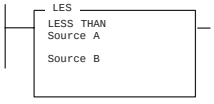
Use the NEQ instruction to test whether two values are not equal. If source A and source B are not equal, the instruction is logically true. If the two values are equal, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Execution Times (µsec) when:

True	False
21.52	6.60

Less Than (LES)



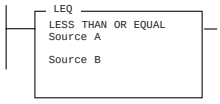
Use the LES instruction to test whether one value (source A) is less than another (source B). If the value at source A is less than the value of source B the instruction is logically true. If the value at source A is greater than or equal to the value of source B, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Execution Times (µsec) when:

True	False
23.60	6.60

Less Than or Equal (LEQ)



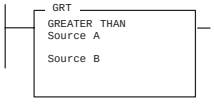
Execution Times (µsec) when:

True	False
23.60	6.60

Use the LEQ instruction to test whether one value (source A) is less than or equal to another (source B). If the value at source A is less than or equal to the value of source B, the instruction is logically true. If the value at source A is greater than the value of source B, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Greater Than (GRT)



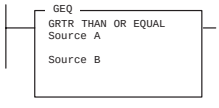
Execution Times (µsec) when:

True	False
23.60	6.60

Use the GRT instruction to test whether one value (source A) is greater than another (source B). If the value at source A is greater than the value of source B, the instruction is logically true. If the value at source A is less than or equal to the value of source B, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Greater Than or Equal (GEQ)



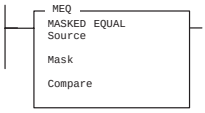
Execution Times (µsec) when:

True	False
23.60	6.60

Use the GEQ instruction to test whether one value (source A) is greater than or equal to another (source B). If the value at source A is greater than or equal to the value of source B, the instruction is logically true. If the value at source A is less than the value of source B, the instruction is logically false.

Source A must be a word address. Source B can be either a constant or word address. Negative integers are stored in two's complement form.

Masked Comparison for Equal (MEQ)



Use the MEQ instruction to compare data of a source address with data of a reference address. Use of this instruction allows portions of the data to be masked by a separate word.

Execution Times (µsec) when:

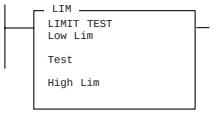
True	False
28.39	7.69

Entering Parameters

- **Source** is the address of the value you want to compare.
- **Mask** is the address of the mask through which the instruction moves data. The mask can be a hexadecimal value (constant).
- **Compare** is an integer value or the address of the reference.

If the 16 bits of data at the source address are equal to the 16 bits of data at the compare address (less masked bits), the instruction is true. The instruction becomes false as soon as it detects a mismatch. Bits in the mask word mask data when reset; they pass data when set.

Limit Test (LIM)



Execution Times (µsec) when:

True	False
36.93	7.69

Use the LIM instruction to test for values within or outside a specified range, depending on how you set the limits.

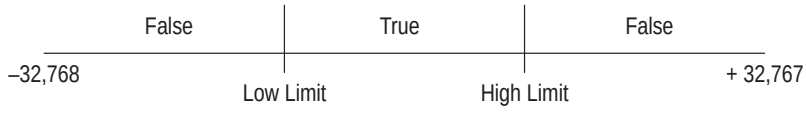
Entering Parameters

The Low Limit, Test, and High Limit values can be word addresses or constants, restricted to the following combinations:

- If the Test parameter is a constant, both the Low Limit and High Limit parameters must be word addresses.
- If the Test parameter is a word address, the Low Limit and High Limit parameters can be either a constant or a word address.

True/False Status of the Instruction

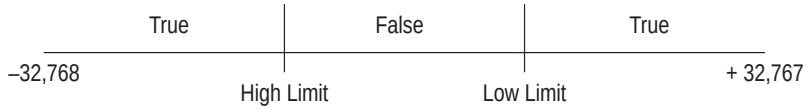
If the Low Limit has a value equal to or less than the High Limit, the instruction is true when the Test value is between the limits or is equal to either limit. If the Test value is outside the limits, the instruction is false, as shown below.



Example, low limit less than high limit:

Low Limit	High Limit	Instruction is True when Test value is	Instruction is False when Test value is
5	8	5 through 8	–32,768 through 4 and 9 through 32,767

If the Low Limit has a value greater than the High Limit, the instruction is false when the Test value is between the limits. If the Test value is equal to either limit or outside the limits, the instruction is true, as shown below.



Example, low limit greater than high limit:

Low Limit	High Limit	Instruction is True when Test value is	Instruction is False when Test value is
8	5	–32,768 through 5 and 8 through 32,767	6 and 7

Comparison Instructions in the Paper Drilling Machine Application Example

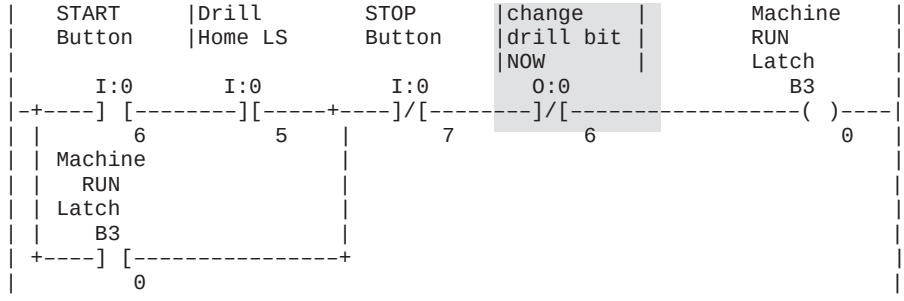
This section provides ladder rungs to demonstrate the use of comparison instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will be adding an instruction to file 2 and beginning a subroutine in file 7.

Adding to File 2

To begin you will need to return to the rungs first entered in chapter 6. One more instruction needs to be added to the first rung to keep track of the drill life. This rung is indicated below by the shading. Notice that text has also been added to the rung comment.

Note *Do not add this instruction if you are using a 16 I/O controller. Address O:0/6 is only valid for 32 I/O controllers.*

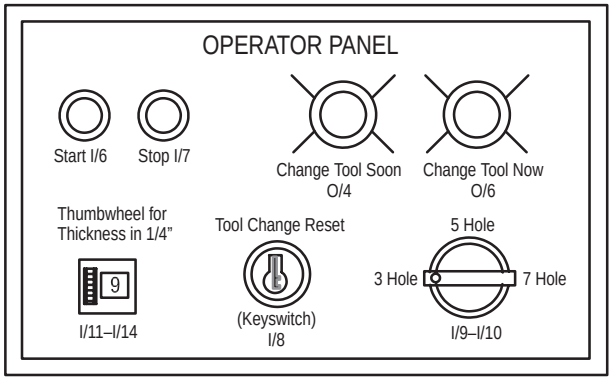
```
Rung 2:3
Starts the conveyor in motion when the start button is pressed.
However, there are other conditions that must also be met before we
start the conveyor.  The are:  the drill must be in its fully
retracted position (home); the drill bit must not be past its maximum
useful life.  This rung also stops the conveyor when the stop button
is pressed or when the drill life is exceeded.
```



Beginning a Subroutine in File 7

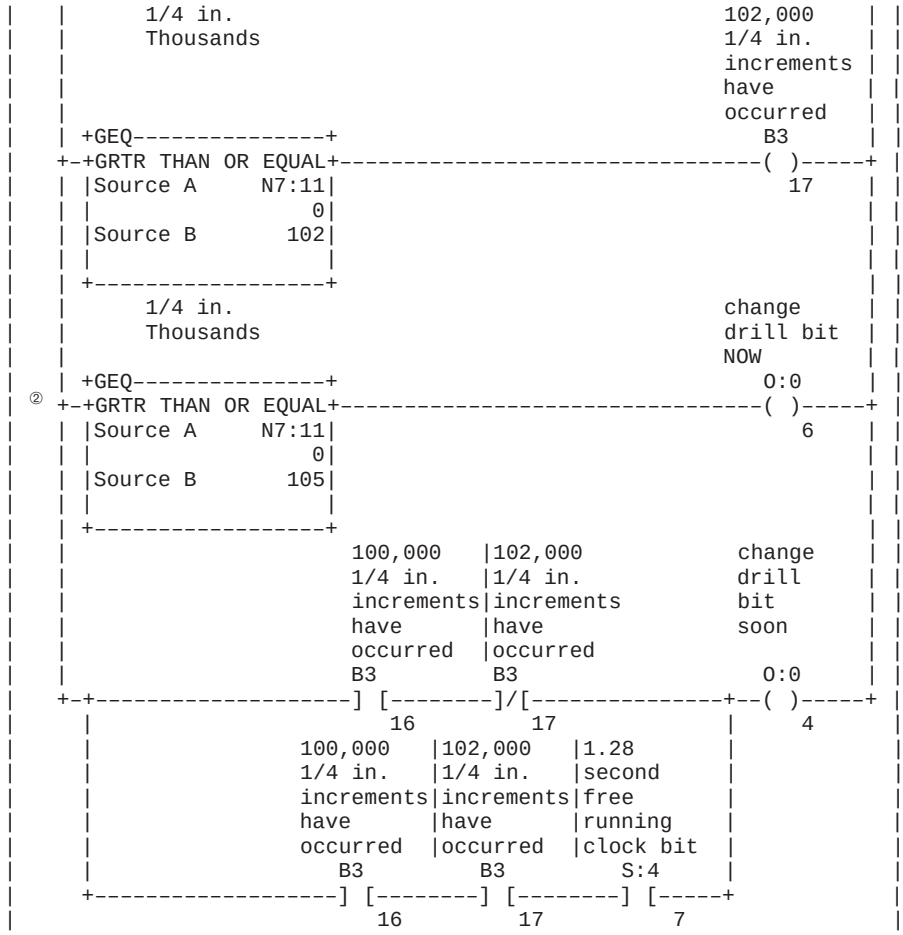
This section of ladder keeps track of the total inches of paper the current drill bit has drilled through. As the current bit wears out, a light illuminates on the operator panel, below, to warn the operator to change the drill bit.

For 32 I/O controllers: If the operator ignores this warning too long, this ladder shuts the machine down until the operator changes the bit.



Rung 7:0①
Examines the number of 1/4 in. thousands that have accumulated over the life of the current drill bit. If the bit has drilled between 100,000–101,999 1/4 in. increments of paper, the ”change drill” light illuminates steadily. When the value is between 102,000–103,999, the ”change drill” light flashes at a 1.28 second rate. When the value reaches 105,000, the ”change drill” light flashes, and the ”change drill now” light illuminates.

```
|          1/4 in.                                    100,000      |
|          Thousands                                  1/4 in.      |
|                                                     increments   |
|                                                     have         |
|                                                     occurred     |
|     +GEQ---------------+                              B3        |
|---+-+GRTR THAN OR EQUAL+-----------------------------( )-----+-|
|   | |Source A     N7:11|                                16    | |
|   | |                0|                                      | |
|   | |Source B      100|                                      | |
|   | |                 |                                      | |
|   | +-----------------+                                      | |
```



① More rungs are added to this subroutine at the end of chapters 8 and 9.

② This branch accesses I/O only available with 32 I/O controllers. Therefore, do not include this branch if you are using a 16 I/O controller.

8 Using Math Instructions

This chapter contains general information about math instructions and explains how they function in your logic program. Each of the math instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the math instructions in use.

Math Instructions

Instruction		Purpose	Page
Mnemonic	Name		
ADD	Add	Adds source A to source B and stores the result in the destination.	**8–4**
SUB	Subtract	Subtracts source B from source A and stores the result in the destination.	**8–5**
MUL	Multiply	Multiplies source A by source B and stores the result in the destination.	**8–8**
DIV	Divide	Divides source A by source B and stores the result in the destination and the math register.	**8–9**
DDV	Double Divide	Divides the contents of the math register by the source and stores the result in the destination and the math register.	**8–10**
CLR	Clear	Sets all bits of a word to zero.	**8–11**
SQR	Square Root	Calculates the square root of the source and places the integer result in the destination.	**8–11**
SCL	Scale Data	Multiplies the source by a specified rate, adds to an offset value, and stores the result in the destination.	**8–12**

About the Math Instructions

These instructions perform the familiar four function math operations. The majority of the instructions take two input values, perform the specified arithmetic function, and output the result to an assigned memory location.

For example, both the ADD and SUB instructions take a pair of input values, add or subtract them, and place the result in the specified destination. If the result of the operation exceeds the allowable value, an overflow or underflow bit is set.

To learn more about the math instructions, we suggest that you read the Math Instructions Overview that follows.

Math Instructions Overview

The following general information applies to math instructions.

Using Indexed Word Addresses

You have the option of using indexed word addresses for instruction parameters specifying word addresses. Indexed addressing is discussed in chapter 5.

Updates to Arithmetic Status Bits

The arithmetic status bits are found in Word 0, bits 0–3 in the controller status file. After an instruction is executed, the arithmetic status bits in the status file are updated:

With this Bit:		The Controller:
S:0/0	Carry (C)	sets if carry is generated; otherwise cleared.
S:0/1	Overflow (V)	indicates that the actual result of a math instruction does not fit in the designated destination.
S:0/2	Zero (Z)	indicates a 0 value after a math, move, or logic instruction.
S:0/3	Sign (S)	indicates a negative (less than 0) value after a math, move, or logic instruction.

Overflow Trap Bit, S:5/0

Minor error bit (S:5/0) is set upon detection of a mathematical overflow or division by zero. If this bit is set upon execution of an END statement or a Temporary End (TND) instruction, the recoverable major error code 0020 is declared.

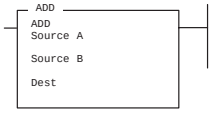
In applications where a math overflow or divide by zero occurs, you can avoid a controller fault by using an unlatch (OTU) instruction with address S:5/0 in your program. The rung must be between the overflow point and the END or TND statement.

Changes to the Math Register, S:13 and S:14

Status word S:13 contains the *least* significant word of the 32-bit values of the MUL and DDV instructions. It contains the remainder for DIV and DDV instructions. It also contains the first four BCD digits for the Convert from BCD (FRD) and Convert to BCD (TOD) instructions.

Status word S:14 contains the *most* significant word of the 32-bit values of the MUL and DDV instructions. It contains the unrounded quotient for DIV and DDV instructions. It also contains the most significant digit (digit 5) for TOD and FRD instructions.

Add (ADD)



Use the ADD instruction to add one value (source A) to another value (source B) and place the result in the destination. Source A and B can either be a word address or constant.

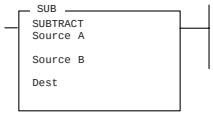
Execution Times (µsec) when:

True	False
33.09	6.78

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	sets if carry is generated; otherwise resets.
S:0/1	Overflow (V)	sets if overflow is detected at destination; otherwise resets. On overflow, the minor error flag is also set. The value –32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets.

Subtract (SUB)



Use the SUB instruction to subtract one value (Source B) from another (source A) and place the result in the destination. Source A and B can either be a word address or constant.

Execution Times (µsec) when:

True	False
33.52	6.78

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	sets if borrow is generated; otherwise resets.
S:0/1	Overflow (V)	sets if underflow; otherwise reset. On underflow, the minor error flag is also set. The value –32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets.

32-Bit Addition and Subtraction

You have the option of performing 16-bit or 32-bit signed integer addition and subtraction. This is facilitated by status file bit S:2/14 (math overflow selection bit).

Math Overflow Selection Bit S:2/14

Set this bit when you intend to use 32-bit addition and subtraction. When S:2/14 is set, and the result of an ADD, SUB, MUL, DIV, or NEG instruction cannot be represented in the destination address (due to math underflow or overflow):

- The overflow bit S:0/1 is set.
- The overflow trap bit S:5/0 is set.
- The destination address contains the unsigned truncated least significant 16 bits of the result.

When S:2/14 is reset (default condition), and the result of an ADD, SUB, MUL, DIV, or NEG instruction cannot be represented in the destination address (due to math underflow or overflow):

- The overflow bit S:0/1 is set.
- The overflow trap bit S:5/0 is set.
- The destination address contains 32767 if the result is positive or –32768 if the result is negative.

Note that the status of bit S:2/14 has no effect on the DDV instruction. Also, it has no effect on the math register content when using MUL and DIV instructions.

Example of 32-bit Addition

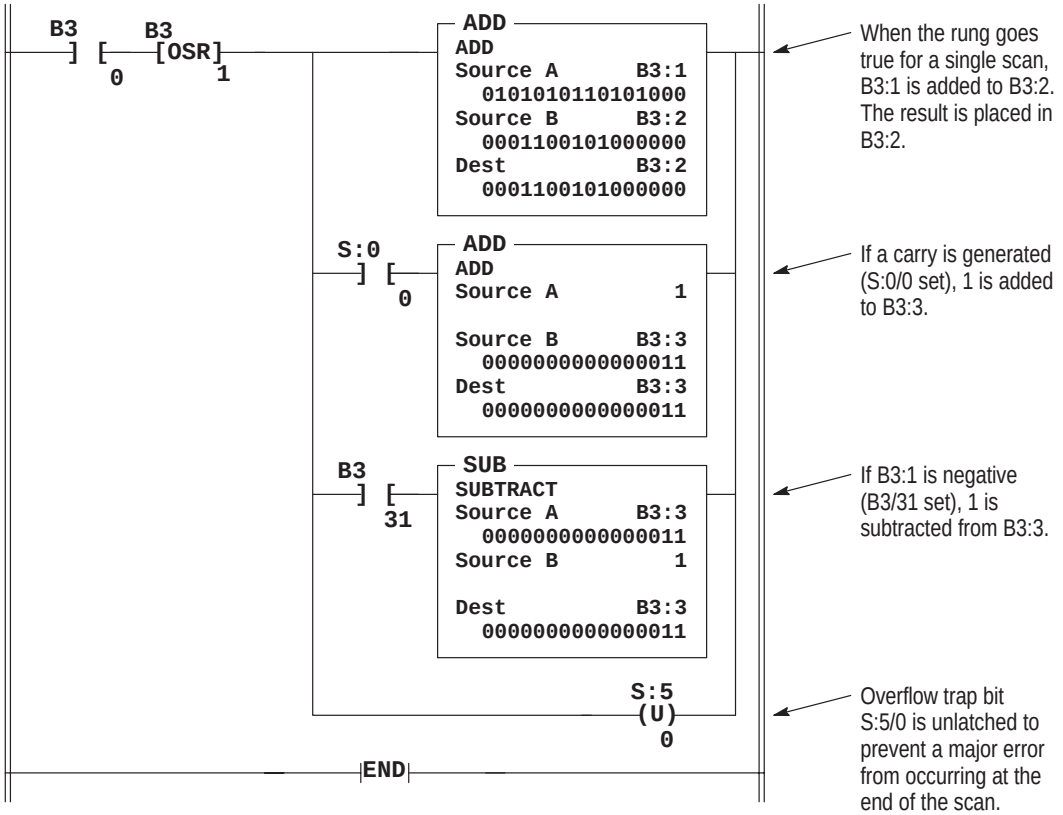
The following example shows how a 16-bit signed integer is added to a 32-bit signed integer. Remember that S:2/14 must be set for 32-bit addition.

Note that the value of the most significant 16 bits (B3:3) of the 32-bit number is increased by 1 if the carry bit S:0/0 is set and it is decreased by 1 if the number being added (B3:1) is negative.

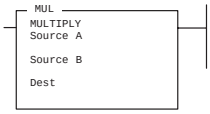
To avoid a major error from occurring at the end of the scan, you must unlatch overflow trap bit S:5/0 as shown.

Add 16–bit value B3:1 to 32–bit value B3:3 B3:2				
Add Operation		Binary	Hex	Decimal①
Addend	B3:3 B3:2	0000 0000 0000 0011 0001 1001 0100 0000	0003 1940	203,072
Addend	B3:1	0101 0101 1010 1000	55A8	21,928
Sum	B3:3 B3:2	0000 0000 0000 0011 0110 1110 1110 1000	0003 6EE8	225,000

① The programming device displays 16-bit decimal values only. The decimal value of a 32-bit integer is derived from the displayed binary or hex value. For example, 0003 1940 Hex is 16^4x3 + 16^3x1 + 16^2x9 + 16^1x4 + 16^0x0 = 203,072.



Multiply (MUL)



Use the MUL instruction to multiply one value (source A) by another (source B) and place the result in the destination. Source A and B can either be a word address or constant.

If the result is larger than +32,767 or smaller than –32,767 (16-bits), the 32-bit result is placed in the math register.

Execution Times (µsec) when:

True	False
57.96	6.78

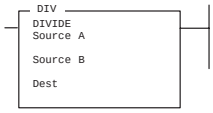
Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if overflow is detected at destination; otherwise resets. On overflow, the minor error flag is also set. The value –32,768 or 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets.

Changes to the Math Register

The math register contains the 32-bit signed integer result of the multiply operation. This result is valid at overflow.

Divide (DIV)



Use the DIV instruction to divide one value (source A) by another (source B), and place the rounded quotient in the destination. If the remainder is 0.5 or greater, the destination is rounded up.

Execution Times (µsec) when:

True	False
147.87	6.78

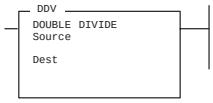
Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if division by zero or overflow is detected; otherwise resets. On overflow, the minor error flag is also set. The value 32,767 is placed in the destination. If S:2/14 (math overflow selection bit) is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets; undefined if overflow is set.
S:0/3	Sign (S)	sets if result is negative; otherwise resets; undefined if overflow is set.

Changes to the Math Register

The unrounded quotient is placed in the most significant word, the remainder is placed in the least significant word.

Double Divide (DDV)



Execution Times (µsec) when:

True	False
157.06	6.78

The 32-bit content of the math register is divided by the 16-bit source value and the rounded quotient is placed in the destination. If the remainder is 0.5 or greater, the destination is rounded up.

This instruction typically follows a MUL instruction that creates a 32-bit result.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if division by zero or if result is greater than 32,767 or less than –32,768; otherwise resets. On overflow, the minor error flag is also set. The value 32,767 is placed in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets; undefined if overflow is set.

Changes to the Math Register

Initially contains the dividend of the DDV operation. Upon instruction execution the unrounded quotient is placed in the most significant word of the math register. The remainder is placed in the least significant word of the math register.

Clear (CLR)

Use the CLR instruction to set the destination to zero. All of the bits reset.

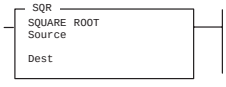
Execution Times (µsec) when:

True	False
20.80	4.25

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	always sets.
S:0/3	Sign (S)	always resets.

Square Root (SQR)



When this instruction is evaluated as true, the square root of the absolute value of the source is calculated and the rounded integer result is placed in the destination.

The instruction calculates the square root of a negative number without overflow or faults. In applications where the source value may be negative, use a comparison instruction to evaluate the source value to determine if the destination may be invalid.

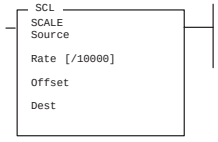
Execution Times (µsec) when:

True	False
71.25	6.78

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	sets if the source is negative; otherwise cleared.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets when destination value is zero.
S:0/3	Sign (S)	always resets.

Scale Data (SCL)



When this instruction is true, the value at the source address is multiplied by the rate value. The rounded result is added to the offset value and placed in the destination.

Execution Times (µsec) when:

True	False
169.18	6.78

Note *Anytime an underflow or overflow occurs in the destination file, minor error bit S:5/0 must be reset. This must occur before the end of the current scan to prevent major error code 0020 from being declared. This instruction can overflow before the offset is added.*

Entering Parameters

The value for the following parameters is between –32,768 to 32,767.

- **Source** can either be a constant or a word address.
- **Rate** is the positive or negative value you enter divided by 10,000. It can be a constant or a word address.
- **Offset** can either be a constant or a word address.

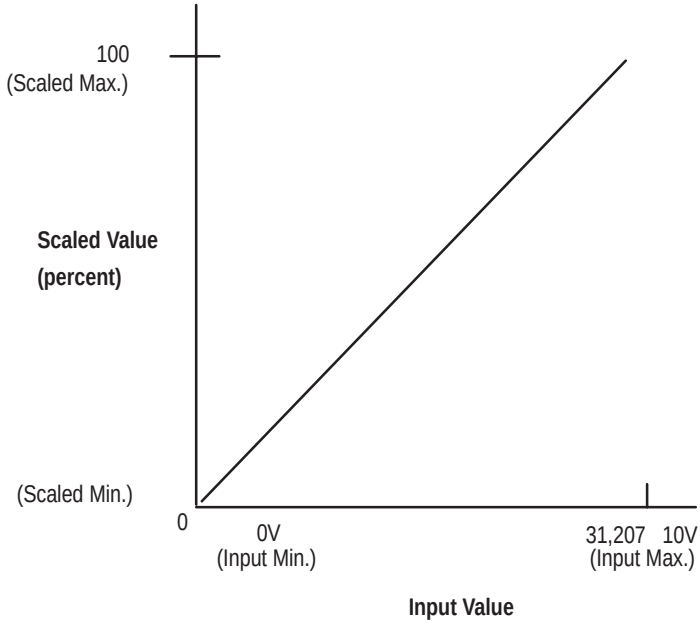
Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	is reserved.
S:0/1	Overflow (V)	sets if an overflow is detected; otherwise resets. On overflow, minor error bit S:5/0 is also set and the value –32,768 or 32,767 is placed in the destination. The presence of an overflow is checked before and after the offset value is applied.①
S:0/2	Zero (Z)	sets when destination value is zero.
S:0/3	Sign (S)	sets if the destination value is negative; otherwise resets.

① If the result of the Source times the Rate, divided by 10000 is greater than 32767, the SCL instruction overflows, causing error 0020 (Minor Error Bit), and places 32767 in the Destination. This occurs regardless of the current offset.

The following example takes a 0V to 10V analog input from a MicroLogix 1000 analog controller and scales the raw input data to a value between 0 and 100%. The input value range is 0V to 10V which corresponds to 0 to 31,207 counts. The scaled value range is 0 to 100 percent.

Application Example – Convert Voltage Input to Percent



Calculating the Linear Relationship

Use the following equations to calculate the scaled units:

Scaled value = (input value x rate) + offset

Rate = (scaled max. – scaled min.) / (input max. – input min.)

= (100 – 0) / (31,207 – 0)

= .00320 (or 32/10000)

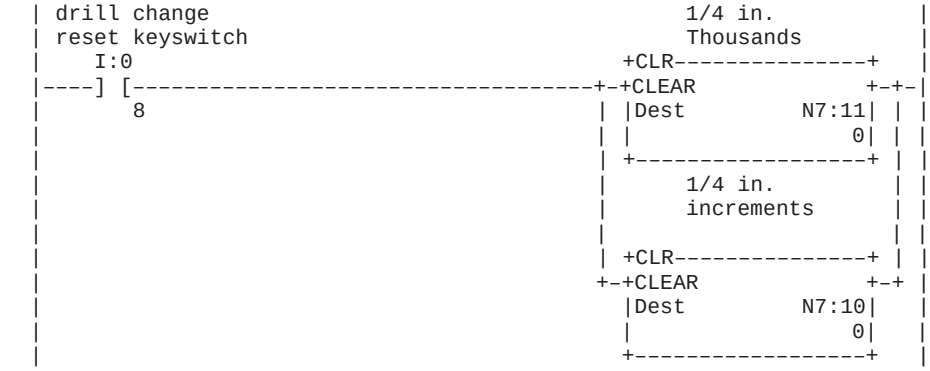
Offset = scaled min. – (input min. x rate)

= 0 – (0 × .00320) = 0

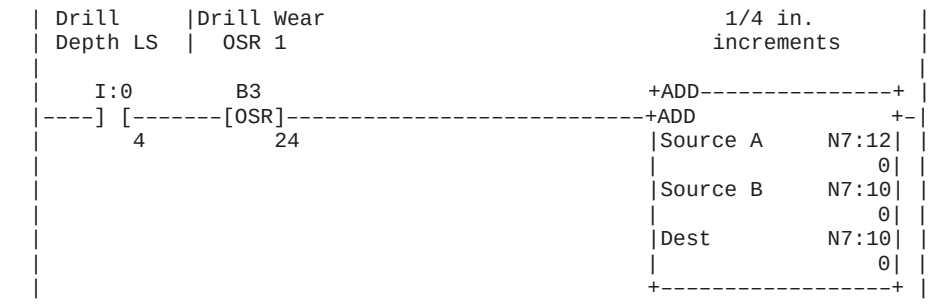
Math Instructions in the Paper Drilling Machine Application Example

This section provides ladder rungs to demonstrate the use of math instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will be adding to the subroutine in file 7 that was started in chapter 7.

Rung 7:1
Resets the number of 1/4 in. increments and the 1/4 in. thousands when the ”drill change reset” keyswitch is energized. This should occur following each drill bit change.

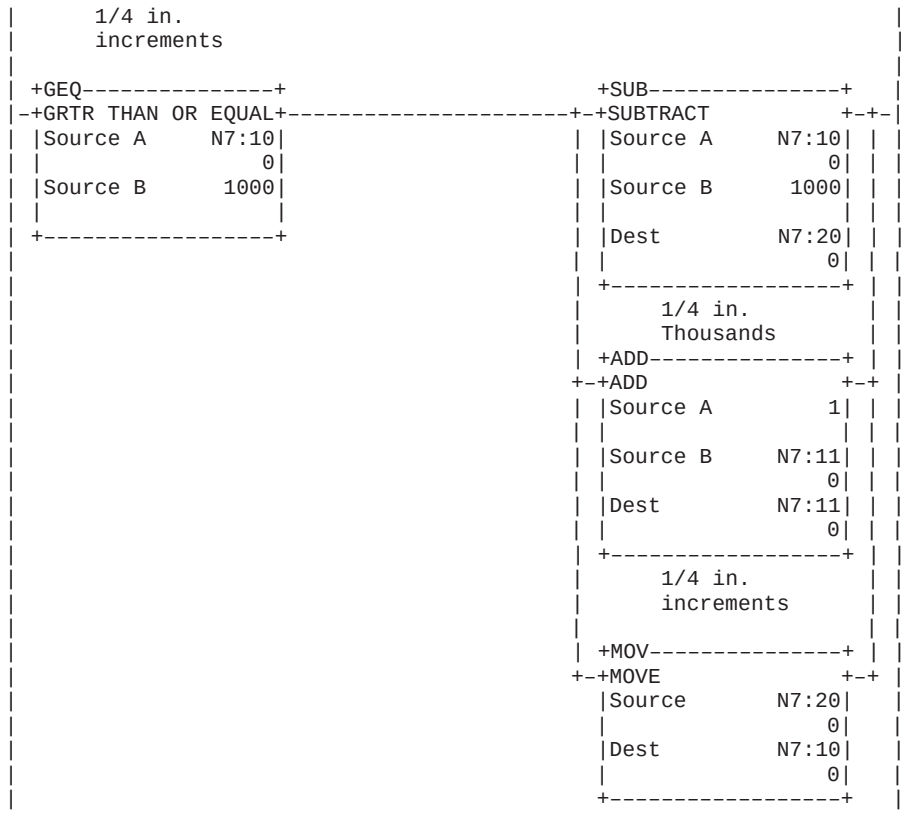


Rung 7:5[1]
Keeps a running total of how many inches of paper have been drilled with the current drill bit. Every time a hole is drilled, the thickness (in 1/4 ins) is added to the running total (kept in 1/4 ins). The OSR is necessary because the ADD executes every time the rung is true, and the drill body would actuate the DRILL DEPTH limit switch for more than 1 program scan. Integer N7:12 is the integer-converted value of the BCD thumbwheel on inputs I:0/11 – I:0/14.

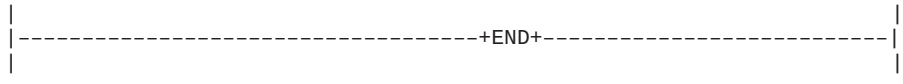


[1] Rungs 7:2 through 7:4 are added at the end of Chapter 9.

Rung 7:6
When the number of 1/4 in. increments surpasses 1000, finds out how many increments are past 1000 and stores in N7:20. Add 1 to the total of 1000 1/4 in. increments, and re-initializes the 1/4 in. increments accumulator to how many increments were beyond 1000.



Rung 7:7



Notes:

9 Using Data Handling Instructions

This chapter contains general information about the data handling instructions and explains how they function in your application program. Each of the instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the data handling instructions in use.

Data Handling Instructions

Instruction		Purpose	Page
Mnemonic	**Name**		
TOD	Convert to BCD	Converts the integer source value to BCD format and stores it in the destination.	**9–3**
FRD	Convert from BCD	Converts the BCD source value to an integer and stores it in the destination.	**9–5**
DCD	Decode 4 to 1 of 16	Decodes a 4-bit value (0 to 15), turning on the corresponding bit in the 16-bit destination.	**9–8**
ENC	Encode 1 of 16 to 4	Encodes a 16-bit source to a 4-bit value. Searches the source from the lowest to the highest bit, and looks for the first set bit. The corresponding bit position is written to the destination as an integer.	**9–9**
COP and FLL	Copy File and Fill File	The COP instruction copies data from the source file to the destination file The FLL instruction loads a source value into each position in the destination file.	**9–10**

Continued on the next page.

Instruction		Purpose	Page
Mnemonic	Name		
MOV	Move	Moves the source value to the destination.	**9–15**
MVM	Masked Move	Moves data from a source location to a selected portion of the destination.	**9–16**
AND	And	Performs a bitwise AND operation.	**9–18**
OR	Or	Performs a bitwise inclusive OR operation.	**9–19**
XOR	Exclusive Or	Performs a bitwise Exclusive OR operation.	**9–20**
NOT	Not	Performs a NOT operation.	**9–21**
NEG	Negate	Changes the sign of the source and stores it in the destination.	**9–22**
FFL and FFU	FIFO Load and FIFO Unload	The FFL instruction loads a word into a FIFO stack on successive false-to-true transitions. The FFU unloads a word from the stack on successive false-true transitions. The first word loaded is the first to be unloaded.	**9–25**
LFL and LFU	LIFO Load and LIFO Unload	The LFL instruction loads a word into a LIFO stack on successive false-to-true transitions. The LFU unloads a word from the stack on successive false-to-true transitions. The last word loaded is the first to be unloaded.	**9–26**

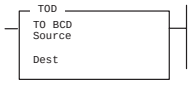
About the Data Handling Instructions

Use these instructions to convert information, manipulate data in the controller, and perform logic operations.

In this chapter you will find a general overview preceding groups of instructions. Before you learn about the instructions in each of these groups, we suggest that you read the overview. This chapter contains the following overviews:

- Move and Logical Instructions Overview
- FIFO and LIFO Instructions Overview

Convert to BCD (TOD)



Execution Times (µsec) when:

True	False
49.64	6.78

Use this instruction to convert 16-bit integers into BCD values.

The source must be a word address. The destination parameter can be a word address in a data file, or it can be the math register, S:13 and S:14.

If the integer value you enter is negative, the sign is ignored and the conversion occurs as if the number was positive.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if the BCD result is larger than 9999. On overflow, the minor error flag is also set.
S:0/2	Zero (Z)	sets if destination value is zero.
S:0/3	Sign (S)	sets if the source word is negative; otherwise resets.

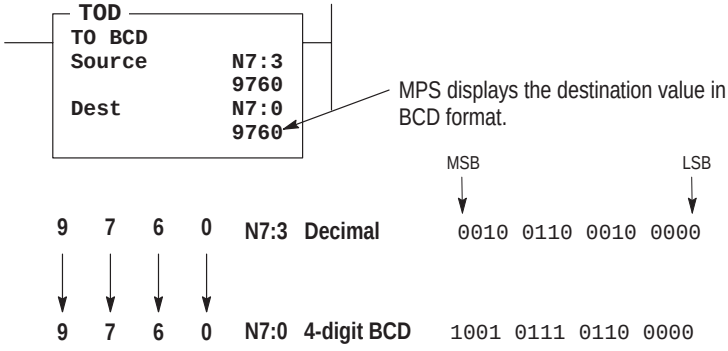
Changes to the Math Register

Contains the 5-digit BCD result of the conversion. This result is valid at overflow.

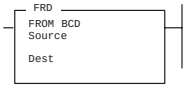
Note *To convert numbers larger than 9999 decimal, the destination must be the Math Register (S:13). You must reset the Minor Error Bit (S:5/0) to prevent an error.*

Example

The integer value 9760 stored at N7:3 is converted to BCD and the BCD equivalent is stored in N7:0. The maximum BCD value is 9999.



Convert from BCD (FRD)



Execution Times (µsec) when:

True	False
56.88	5.52

Use this instruction to convert BCD values to integer values.

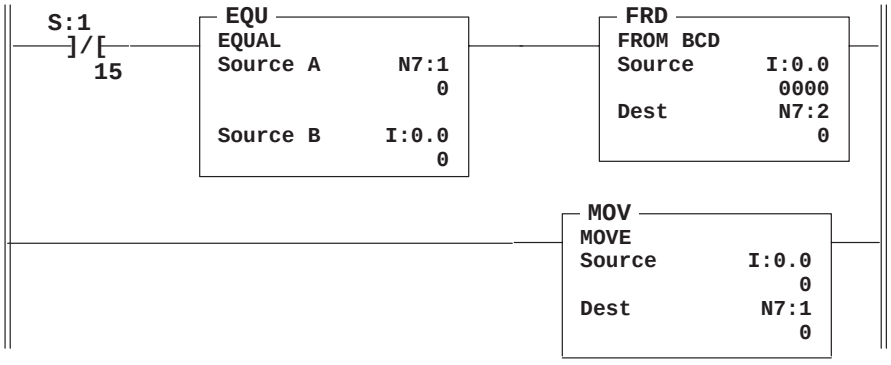
The source parameter can be a word address in a data file, or it can be the math register, S:13. The destination must be a word address.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if non-BCD value is contained at the source or the value to be converted is greater than 32,767; otherwise reset. On overflow, the minor error flag is also set.
S:0/2	Zero (Z)	sets if destination value is zero.
S:0/3	Sign (S)	always resets.

Note

Always provide ladder logic filtering of all BCD input devices prior to performing the FRD instruction. The slightest difference in point-to-point input filter delay can cause the FRD instruction to overflow due to the conversion of a non-BCD digit.

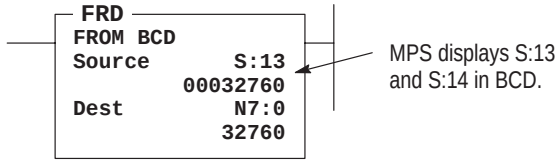


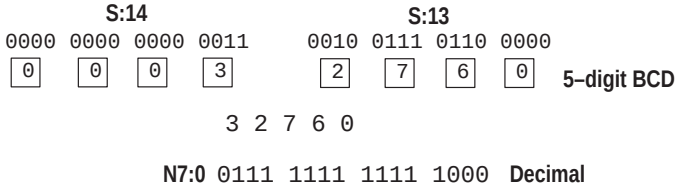
The two rungs shown cause the controller to verify that the value I:0 remains the same for two consecutive scans before it will execute the FRD. This prevents the FRD from converting a non-BCD value during an input value change.

Note *To convert numbers larger than 9999 BCD, the source must be the Math Register (S:13). You must reset the Minor Error Bit (S:5.0) to prevent an error.*

Example

The BCD value 32,760 in the math register is converted and stored in N7:0. The maximum source value is 32767, BCD.

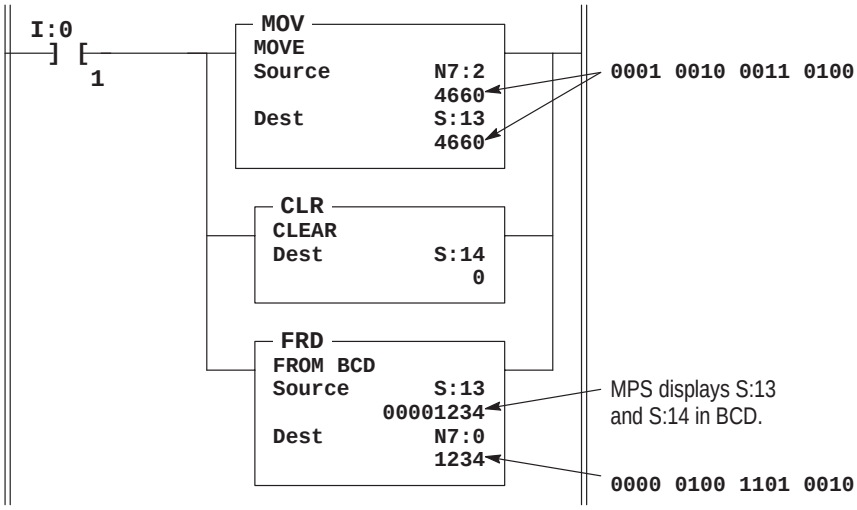




You should convert BCD values to integer before you manipulate them in your ladder program. If you do not convert the values, the controller manipulates them as integers and their value may be lost.

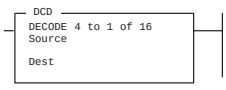
Note *If the math register (S:13 and S:14) is used as the source for the FRD instruction and the BCD value does not exceed 4 digits, be sure to clear word S:14 before executing the FRD instruction. If S:14 is not cleared and a value is contained in this word from another math instruction located elsewhere in the program, an incorrect decimal value will be placed in the destination word.*

Clearing S:14 before executing the FRD instruction is shown below:



When the input condition I:0/1 is set (1), a BCD value (transferred from a 4-digit thumbwheel switch for example) is moved from word N7:2 into the math register. Status word S:14 is then cleared to make certain that unwanted data is not present when the FRD instruction is executed.

Decode 4 to 1 of 16 (DCD)



When executed, this instruction sets one bit of the destination word. The particular bit that is turned On depends on the value of the first four bits of the source word. See the table below.

Use this instruction to multiplex data in applications such as rotary switches, keypads, and bank switching.

Execution Times (µsec) when:

True	False
27.67	6.78

	Source					Destination														
Bit	15–04	03	02	01	00	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	x	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	x	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	x	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	x	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	x	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	x	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	x	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	x	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	x	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	x	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	x	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	x	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	x	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	x	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	x	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

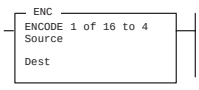
Entering Parameters

- **Source** is the address that contains the information to be decoded. Only the first four bits (0–3) are used by the DCD instruction. The remaining bits may be used for other application specific needs.
- **Destination** is the address of the word where the decoded data is to be stored.

Updates to Arithmetic Status Bits

Unaffected.

Encode 1 of 16 to 4 (ENC)



When the rung is true, this output instruction searches the source from the lowest to the highest bit, and looks for the first set bit. The corresponding bit position is written to the destination as an integer as shown in the table below.

Execution Times (µsec) when:

True	False
54.80	6.78

	Source																Destination				
Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	15–04	03	02	01	00
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	0	0	0	0
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	x	0	0	0	1
	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	x	0	0	1	0
	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	x	0	0	1	1
	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	x	0	1	0	0
	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	x	0	1	0	1
	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	x	0	1	1	0
	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	x	0	1	1	1
	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	x	1	0	0	0
	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	x	1	0	0	1
	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	x	1	0	1	0
	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	x	1	0	1	1
	x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	x	1	1	0	0
	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	x	1	1	0	1
	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	1	1	1	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	1	1	1	1

Entering Parameters

- **Source** is the address of the word to be encoded. Only one bit of this word should be on at any one time. If more than one bit in the source is set, the destination bits will be set based on the least significant bit that is set. If a source of zero is used, all of the destination bits will be reset and the zero bit will be set.
- **Destination** is the address that contains the bit encode information. Bits 4–15 of the destination are reset by the ENC instruction.

Updates to Arithmetic Status Bits

The arithmetic status bits are found in Word 0, bits 0–3 in the controller status file. After an instruction is executed, the arithmetic status bits in the status file are updated:

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	sets if more than one bit in the source is set; otherwise reset. The math overflow bit (S:5/0) is *not* set.
S:0/2	Zero (Z)	sets if destination value is zero.
S:0/3	Sign (S)	always resets.

Copy File (COP) and Fill File (FLL) Instructions

The destination file type determines the number of words that an instruction transfers. For example, if the destination file type is a counter and the source file type is an integer, three integer words are transferred for each element in the counter-type file.

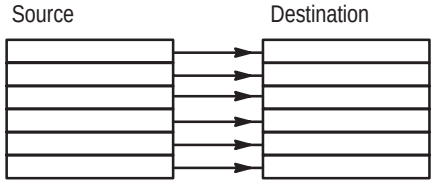
After a COP or FLL instruction is executed, index register S:24 is cleared to zero.

Execution Times (µsec) when:

	True	False
COP	2731+5.06/word	7
FLL	26.86+3.62/word	7

Using COP

This instruction copies blocks of data from one location into another. It uses no status bits. If you need an enable bit, program an output instruction (OTE) in parallel using an internal bit as the output address. The following figure shows how file instruction data is manipulated.



File to File

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** is the address of the first word in the file to be copied. You must use the file indicator (#) in the address.
- **Destination** is the address of the first word in the file where the data is to be stored. You must use the file indicator (#) in the address.
- **Length** is the number of *words or elements* in the file to be copied. See the table on the next page.

If the destination file type is a(n):	then you can specify a maximum length of:	
	Discrete	Analog
Output	1	5
Input	2	8
Status	33	33
Bit	32	32
Timer	40	40
Counter	32	32
Control	16	16
Integer	105	105

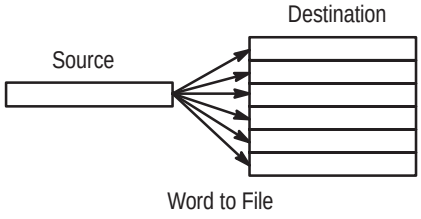
Note *The maximum lengths apply when the source is of the same file type.*

All elements are copied from the source file into the destination file each time the instruction is executed. Elements are copied in ascending order.

If your destination file type is a timer, counter, or control file, be sure the source words corresponding to the status elements of your destination file contain zeros.

Using FLL

The following figure shows how file instruction data is manipulated. The instruction fills the words of a file with a source value. It uses no status bits. If you need an enable bit, program a parallel output that uses a storage address.



Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** is a constant or element address. The file indicator (#) is *not* required for an element address.
- **Destination** is the starting address of the file you want to fill. You must use the file indicator (#) in the address.
- **Length** is the number of *words or elements* in the file to be filled.

If the destination file type is a:	then you can specify a maximum length of:	
	Discrete	Analog
Output	1	5
Input	2	8
Status	33	33
Bit	32	32
Timer	40	40
Counter	32	32
Control	16	16
Integer	105	105

All elements are filled from the source value (typically a constant) into the specified destination file each scan the rung is true. Elements are filled in ascending order.

Move and Logical Instructions Overview

The following general information applies to move and logical instructions.

Entering Parameters

- **Source** is the address of the value on which the logical or move operation is to be performed. It can be a word address or a constant. If the instruction has two source operands, it will not accept constants in both operands.
- **Destination** is the address where the resulting data is stored. It must be a word address.

Using Indexed Word Addresses

You have the option of using indexed word addresses for instruction parameters specifying word addresses. Indexed addressing is discussed in chapter 4.

Updates to Arithmetic Status Bits

The arithmetic status bits are found in Word 0, bits 0–3 in the controller status file. After an instruction is executed, the arithmetic status bits in the status file are updated:

Bit	Name	Description
S:0/0	Carry (C)	Set if a carry is generated; otherwise cleared.
S:0/1	Overflow (V)	Indicates that the actual result of a math instruction does not fit in the designated destination.
S:0/2	Zero (Z)	Indicates a 0 value after a math, move, or logic instruction.
S:0/3	Sign (S)	Indicates a negative (less than 0) value after a math, move, or logic instruction.

Overflow Trap Bit, S:5/0

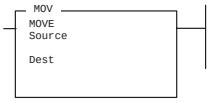
Minor error bit (S:5/0) is set upon detection of a mathematical overflow or division by zero. If this bit is set upon execution of an END statement, or a TND instruction, a major error occurs.

In applications where a math overflow or divide by zero occurs, you can avoid a controller fault by using an unlatch (OTU) instruction with address S:5/0 in your program. The rung must be between the overflow point and the END or TND statement.

Changes to the Math Register, S:13 and S:14

Move and logical instructions do not affect the math register.

Move (MOV)



This output instruction moves the source data to the destination location. As long as the rung remains true, the instruction moves the data each scan.

Execution Times (µsec) when:

True	False
25.05	6.78

Entering Parameters

Enter the following parameters when programming this instruction:

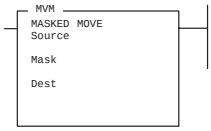
- **Source** is the address or constant of the data you want to move.
- **Destination** is the address where the instruction moves the data.

If you wish to move one word of data without affecting the math flags, use a copy (COP) instruction with a length of 1 word instead of the MOV instruction.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative (most significant bit is set); otherwise resets.

Masked Move (MVM)



The MVM instruction is a word instruction that moves data from a source location to a destination, and allows portions of the destination data to be masked by a separate word. As long as the rung remains true, the instruction moves the data each scan.

Execution Times (µsec) when:

True	False
33.28	6.78

Entering Parameters

Enter the following parameters when programming this instruction:

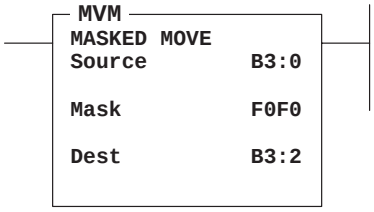
- **Source** is the address of the data you want to move.
- **Mask** is the address of the mask through which the instruction moves data; the mask can be a hex value (constant).
- **Destination** is the address where the instruction moves the data.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets.

Operation

When the rung containing this instruction is true, data at the source address passes through the mask to the destination address. See the following figure.



B3:2 before move

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

source B3:0

0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Mask F0F0

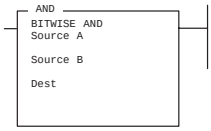
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

B3:2 after move

0	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1

Mask data by setting bits in the mask to zero; pass data by setting bits in the mask to one. The mask can be a constant value, or you can vary the mask by assigning a direct address. *Bits in the destination that correspond to zeros in the mask are not altered.*

And (AND)



Execution Times (µsec) when:

True	False
34.00	6.78

The value at source A is ANDed bit by bit with the value at source B and then stored in the destination.

Truth Table

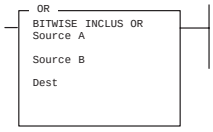
Dest = A AND B		
A	**B**	**Dest**
0	0	0
1	0	0
0	1	0
1	1	1

Source A and B can either be a word address or a constant; however, both sources cannot be a constant. The destination must be a word address.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if most significant bit is set; otherwise resets.

Or (OR)



Execution Times (µsec) when:

True	False
33.68	6.78

The value at source A is ORed bit by bit with the value at source B and then stored in the destination.

Truth Table

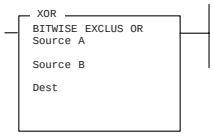
Dest = A OR B		
A	**B**	**Dest**
0	0	0
1	0	1
0	1	1
1	1	1

Source A and B can either be a word address or a constant; however, both sources cannot be a constant. The destination must be a word address.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative (most significant bit is set) otherwise resets.

Exclusive Or (XOR)



Execution Times (µsec) when:

True	False
33.64	6.92

The value at source A is Exclusive ORed bit by bit with the value at source B and then stored in the destination.

Truth Table

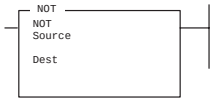
Dest = A XOR B		
A	**B**	**Dest**
0	0	0
1	0	1
0	1	1
1	1	0

Source A and B can either be a word address or a constant; however, both sources cannot be a constant. The destination must be a word address.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets
S:0/3	Sign (S)	sets if result is negative (most significant bit is set); otherwise resets.

Not (NOT)



The source value is NOTed bit by bit and then stored in the destination (one's complement).

Execution Times (µsec) when:

True	False
28.21	6.92

Truth Table

Dest = NOT A	
A	**Dest**
0	1
1	0

The source and destination must be word addresses.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	always resets.
S:0/1	Overflow (V)	always resets.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative (most significant bit is set); otherwise resets.

Negate (NEG)

Execution Times (µsec) when:

True	False
29.48	6.78

Use the NEG instruction to change the sign of a value. If you negate a negative value, the result is a positive; if you negate a positive value, the result is a negative. The destination contains the two's complement of the source.

The source and destination must be word addresses.

Updates to Arithmetic Status Bits

With this Bit:		The Controller:
S:0/0	Carry (C)	clears if 0 or overflow, otherwise sets.
S:0/1	Overflow (V)	sets if overflow, otherwise reset. Overflow occurs only if –32,768 is the source. On overflow, the minor error flag is also set. The value 32,767 is placed in the destination. If S:2/14 is set, then the unsigned, truncated overflow remains in the destination.
S:0/2	Zero (Z)	sets if result is zero; otherwise resets.
S:0/3	Sign (S)	sets if result is negative; otherwise resets.

FIFO and LIFO Instructions Overview

FIFO instructions load words into a file and unload them in the same order as they were loaded. The first word in is the first word out.

LIFO instructions load words into a file and unload them in the opposite order as they were loaded. The last word in is the first word out.

Entering Parameters

Enter the following parameters when programming these instructions:

- **Source** is a word address or constant (–32,768 to 32,767) that becomes the next value in the stack.
- **Destination** (Dest) is a word address that stores the value that exits from the stack.

This Instruction:	Unloads the Value from:
FIFO's FFU	First word
LIFO's LFU	The last word entered

- **FIFO/LIFO** is the address of the stack. It must be an indexed word address in the bit, input, output, or integer file. Use the same FIFO address for the associated FFL and FFU instructions; use the same LIFO address for the associated LFL and LFU instructions.
- **Length** specifies the maximum number of words in the stack. Address the length value by mnemonic (LEN).
- **Position** is the next available location where the instruction loads data into the stack. This value changes after each load or unload operation. Address the position value by mnemonic (POS).

- **Control** is the address of the control structure. The control structure stores the status bits, the stack length, and the position value. Do not use the control file address for any other instruction.

  Status bits of the control structure are addressed by mnemonic. These include:

  - **Empty Bit EM** (bit 12) is set by the controller to indicate the stack is empty.
  - **Done Bit DN** (bit 13) is set by the controller to indicate the stack is full. This inhibits loading the stack.
  - **FFU/LFU Enable Bit EU** (bit 14) is set on a false-to-true transition of the FFU/LFU rung and is reset on a true-to-false transition.
  - **FFL/LFL Enable Bit EN** (bit 15) is set on a false-to-true transition of the FFL/LFL rung and is reset on a true-to-false transition.

Effects on Index Register S:24

The value present in S:24 is overwritten with the position value when a false-to-true transition of the FFL/FFU or LFL/LFU rung occurs. For the FFL/LFL, the position value determined at instruction entry is placed in S:24. For the FFU/LFU, the position value determined at instruction exit is placed in S:24.

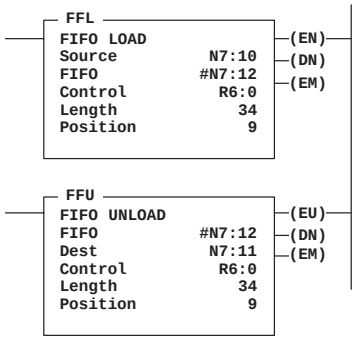
When the DN bit is set, a false-to-true transition of the FFL/LFL rung does not change the position value or the index register value. When the EM bit is set, a false-to-true transition of the FFU/LFU rung does not change the position value or the index register value.

FIFO Load (FFL) and FIFO Unload (FFU)

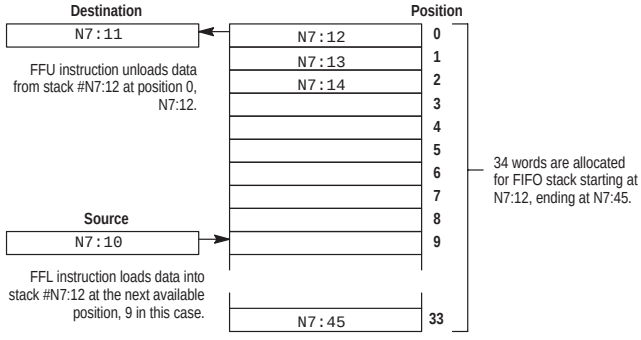
FFL and FFU instructions are used in pairs. The FFL instruction loads words into a user-created file called a FIFO stack. The FFU instruction unloads words from the FIFO stack in the same order as they were entered.

Operation

Instruction parameters have been programmed in the FFL – FFU instruction pair shown below.



FFL–FFU Instruction Pair



Loading and Unloading of Stack #N7:12

FFL Instruction

Execution Times (µsec) when:

True	False
61.13	33.67

When rung conditions change from false-to-true, the controller sets the FFL enable bit (EN). This loads the contents of the Source, N7:10, into the stack structure indicated by the Position number, 9. The position value then increments.

The FFL instruction loads an element at each false-to-true transition of the rung, until the stack is filled (34 elements). The controller then sets the done bit (DN), inhibiting further loading.

FFU Instruction

Execution Times (µsec) when:	
True	False
73.78+ 4.34/word	34.90

When rung conditions change from false-to-true, the controller sets the FFU enable bit (EU). This unloads the contents of the element at stack position 0 into the Destination, N7:11. All data in the stack is shifted one element toward position zero, and the highest numbered element is zeroed. The position value then decrements.

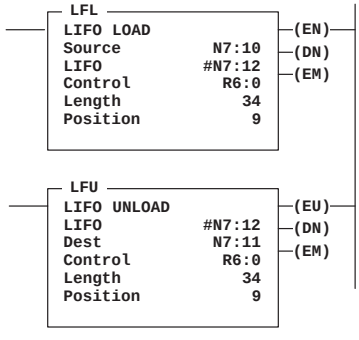
The FFU instruction unloads an element at each false-to-true transition of the rung, until the stack is empty. The controller then sets the empty bit (EM).

LIFO Load (LFL) and LIFO Unload (LFU)

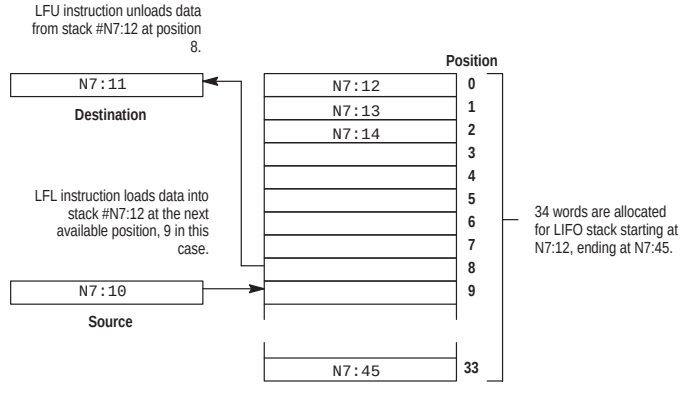
LFL and LFU instructions are used in pairs. The LFL instruction loads words into a user-created file called a LIFO stack. The LFU instruction unloads words from the LIFO stack in the opposite order as they were entered.

Operation

Instruction parameters have been programmed in the LFL – LFU instruction pair shown below.



LFL–LFU Instruction Pair



Loading and Unloading of Stack #N7:12

LFL Instruction

Execution Times (µsec) when:

True	False
61.13	33.67

When rung conditions change from false-to-true, the controller sets the LFL enable bit (EN). This loads the contents of the Source, N7:10, into the stack element indicated by the Position number, 9. The position value then increments.

The LFL instruction loads an element at each false-to-true transition of the rung, until the stack is filled (34 elements). The controller sets the done bit (DN), inhibiting further loading.

LFU Instruction

Execution Times (µsec) when:

True	False
64.20	35.08

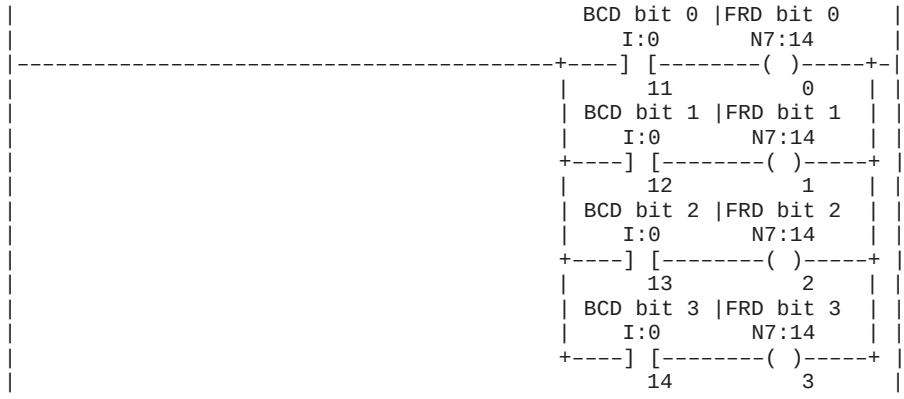
When rung conditions change from false-to-true, the controller sets the LFU enable bit (EU). This unloads data from the last element loaded into the stack (at the position value minus 1), placing it in the Destination, N7:11. The position value then decrements.

The LFU instruction unloads one element at each false-to-true transition of the rung, until the stack is empty. The controller then sets the empty bit (EM).

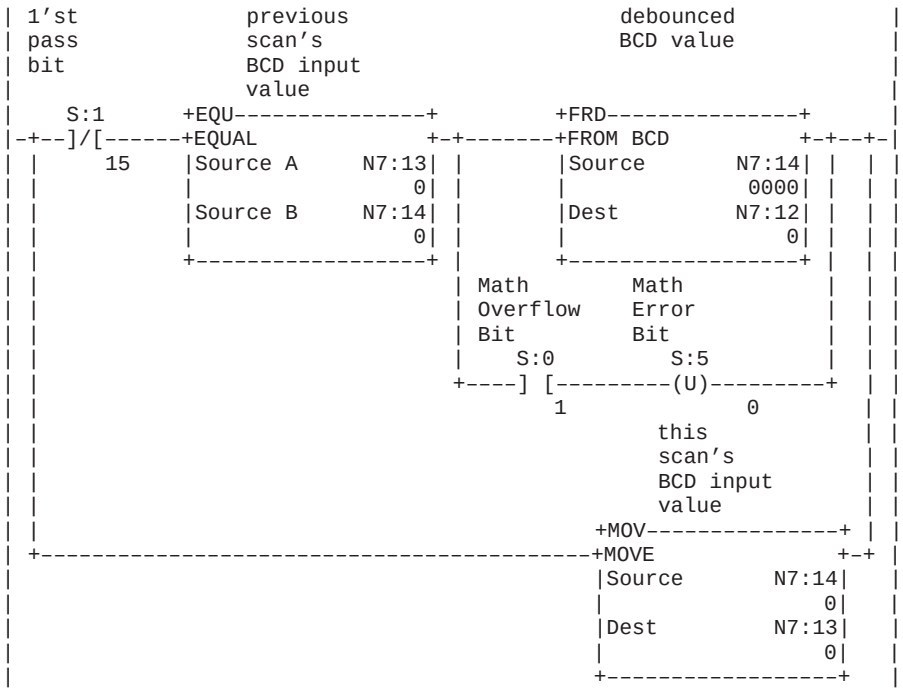
Data Handling Instructions in the Paper Drilling Machine Application Example

This section provides ladder rungs to demonstrate the use of data handling instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will be adding to the subroutine in file 7 that was started in chapter 7.

Rung 7:2①
Moves the single digit BCD thumbwheel value into an internal integer register. This is done to properly align the four BCD input signals prior to executing the BCD to Integer instruction (FRD). The thumbwheel is used to allow the operator to enter the thickness of the paper that is to be drilled. The thickness is entered in 1/4 in. increments. This provides a range of 1/4 in. to 2.25 in.

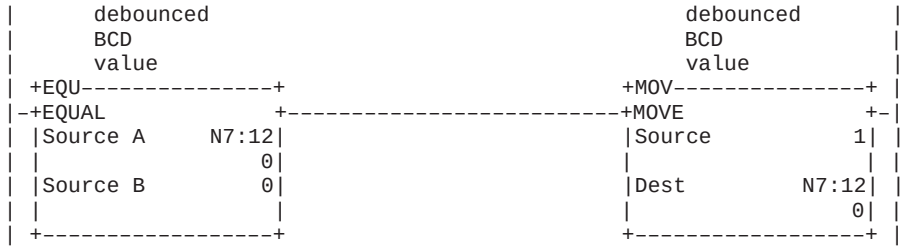


Rung 7:3
Converts the BCD thumbwheel value from BCD to integer. This is done because the controller operates upon integer values. This rung also ”debounces” the thumbwheel to ensure that the conversion only occurs on valid BCD values. Note that invalid BCD values can occur while the operator is changing the BCD thumbwheel. This is due to input filter propagation delay differences between the 4 input circuits that provide the BCD input value.



① This rung accesses I/O only available with 32 I/O controllers. Therefore, do not include this rung if you are using a 16 I/O controller.

Rung 7:4
Ensures that the operator cannot select a paper thickness of 0. If this were allowed, the drill bit life calculation could be defeated resulting in poor quality holes due to a dull drill bit. Therefore the minimum paper thickness used to calculate drill bit wear is 1/4 in.



10 Using Program Flow Control Instructions

This chapter contains general information about the program flow instructions and explains how they function in your application program. Each of the instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the program flow control instructions in use.

Program Flow Control Instructions

Instruction		Purpose	Page
Mnemonic	**Name**		
JMP and LBL	Jump to Label and Label	Jump forward or backward to the specified label instruction.	**10–2**
JSR, SBR, and RET	Jump to Subroutine, Subroutine, and Return from Subroutine	Jump to a designated subroutine and return.	**10–4**
MCR	Master Control Reset	Turn off all non-retentive outputs in a section of ladder program.	**10–7**
TND	Temporary End	Mark a temporary end that halts program execution.	**10–8**
SUS	Suspend	Identifies specific conditions for program debugging and system troubleshooting.	**10–8**
IIM	Immediate Input with Mask	Program an Immediate Input with Mask.	**10–9**
IOM	Immediate Output with Mask	Program an Immediate Output with Mask.	**10–9**

About the Program Flow Control Instructions

Use these instructions to control the sequence in which your program is executed.

Jump (JMP) and Label (LBL)

—(JMP)—

——] LBL [——

Execution Times (µsec) when:

	True	False
JMP	9.04	6.78
LBL	1.45	0.99

Use these instructions in pairs to skip portions of the ladder program.

If the Rung Containing the Jump Instruction is:	Then the Program:
True	Skips from the rung containing the JMP instruction to the rung containing the designated LBL instruction and continues executing. You can jump forward or backward.
False	Does not execute the JMP instruction.

Jumping forward to a label saves program scan time by omitting a program segment until needed. Jumping backward lets the controller execute program segments repeatedly.

Note *Be careful not to jump backwards an excessive number of times. The watchdog timer could time out and fault the controller. Use a counter, timer, or the "program scan" register (system status register, word S:3, bits 0–7) to limit the amount of time you spend looping inside of JMP/LBL instructions.*

Entering Parameters

Enter a decimal label number from 0 to 999. You can place up to 1,000 labels in each subroutine file.

Using JMP

The JMP instruction causes the controller to skip rungs. You can jump to the same label from one or more JMP instruction.

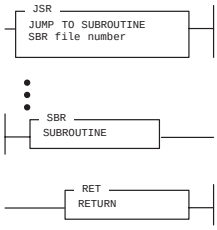
Using LBL

This input instruction is the target of JMP instructions having the same label number. You must program this instruction as the first instruction of a rung. This instruction has no control bits.

You can program multiple jumps to the same label by assigning the same label number to multiple JMP instructions. However, label numbers must be unique.

Note *Do not jump (JMP) into an MCR zone. Instructions that are programmed within the MCR zone starting at the LBL instruction and ending at the 'END MCR' instruction will always be evaluated as though the MCR zone is true, regardless of the true state of the "Start MCR" instruction.*

Jump to Subroutine (JSR), Subroutine (SBR), and Return (RET)



The JSR, SBR, and RET instructions are used to direct the controller to execute a separate subroutine file within the ladder program and return to the instruction following the JSR instruction.

Execution Times (µsec) when:

	True	False
JSR	22.24	4.25
SBR	1.45	0.99
RET	31.11	3.16

Note *If you use the SBR instruction, the SBR instruction must be the first instruction on the first rung in the program file that contains the subroutine.*

Use a subroutine to store recurring sections of program logic that must be executed from several points within your application program. A subroutine saves memory because you program it only once.

Update critical I/O within subroutines using immediate input and/or output instructions (IIM, IOM), especially if your application calls for nested or relatively long subroutines. Otherwise, the controller does not update I/O until it reaches the end of the main program (after executing all subroutines).

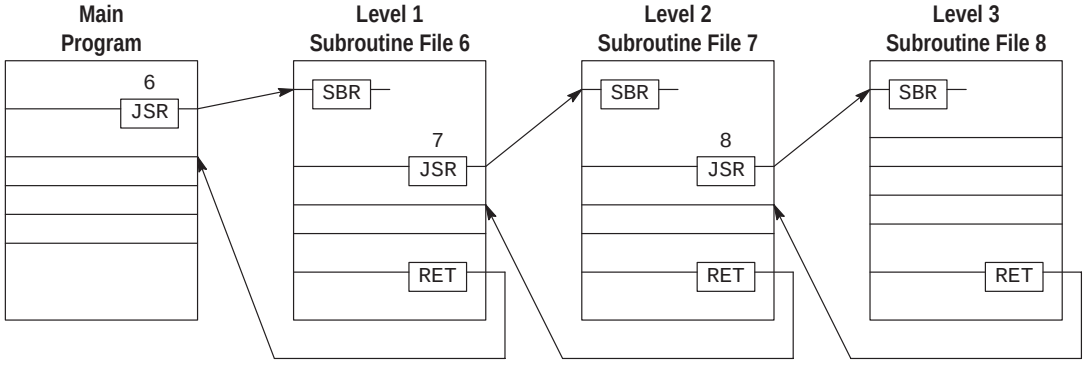
Outputs controlled within a subroutine remain in their last state until the subroutine is executed again.

Nesting Subroutine Files

Nesting subroutines allows you to direct program flow from the main program to a subroutine and then on to another subroutine.

You can nest up to eight levels of subroutines. If you are using an STI subroutine, HSC interrupt subroutine, or user fault routine, you can nest subroutines up to three levels from each subroutine.

The following figure illustrates how subroutines may be nested.



Example of Nesting Subroutines to Level 3

An error occurs if more than the allowable levels of subroutines are called (subroutine stack overflow) or if more returns are executed than there are call levels (subroutine stack underflow).

Using JSR

When the JSR instruction is executed, the controller jumps to the subroutine instruction (SBR) at the beginning of the target subroutine file and resumes execution at that point. You cannot jump into any part of a subroutine except the first instruction in that file.

You must program each subroutine in its own program file by assigning a unique file number (4–15).

Using SBR

The target subroutine is identified by the file number that you entered in the JSR instruction. This instruction serves as a label or identifier for a program file as a regular subroutine file.

This instruction has no control bits. It is always evaluated as true. The instruction must be programmed as the first instruction of the first rung of a subroutine. Use of this instruction is optional; however, we recommend using it for clarity.

Using RET

This output instruction marks the end of subroutine execution or the end of the subroutine file. It causes the controller to resume execution at the instruction following the JSR instruction.

The rung containing the RET instruction may be conditional if this rung precedes the end of the subroutine. In this way, the controller omits the balance of a subroutine only if its rung condition is true.

Without an RET instruction, the END instruction (always present in the subroutine) automatically returns program execution to the instruction following the JSR instruction in your calling ladder file.

Master Control Reset (MCR)

—(MCR)—

Execution Times (µsec) when:

True	False
3.98	4.07

Use MCR instructions in pairs to create program zones that turn off all the non-retentive outputs in the zone. Rungs within the MCR zone are still scanned, but scan time is reduced due to the false state of non-retentive outputs. Non-retentive outputs are reset when their rung goes false.

If the MCR Rung that Starts the Zone is:	Then the Controller:
True	Executes the rungs in the MCR zone based on each rung's individual input condition (as if the zone did not exist).
False	Resets all non-retentive output instructions in the MCR zone regardless of each rung's individual input conditions.

MCR zones let you enable or inhibit segments of your program, such as for recipe applications.

When you program MCR instructions, note that:

- You must end the zone with an unconditional MCR instruction.
- You cannot nest one MCR zone within another.
- Do not jump into an MCR zone. If the zone is false, jumping into it activates the zone.

Note

The MCR instruction is not a substitute for a hard-wired master control relay that provides emergency stop capability. You still must install a hard-wired master control relay to provide emergency I/O power shutdown.

If you start instructions such as timers or counters in an MCR zone, instruction operation ceases when the zone is disabled. Re-program critical operations outside the zone if necessary.

Temporary End (TND)

—(TND)—

Execution Times (µsec) when:

True	False
7.78	3.16

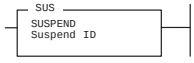
This instruction, when its rung is true, stops the controller from scanning the rest of the program file, updates the I/O, and resumes scanning at rung 0 of the main program (file 2). If this instruction's rung is false, the controller continues the scan until the next TND instruction or the END statement. Use this instruction to progressively debug a program, or conditionally omit the balance of your current program file or subroutines.

Note

If you use this instruction inside a nested subroutine, execution of all nested subroutines is terminated.

Do not execute this instruction from the user error fault routine (file 3), high-speed counter interrupt routine (file 4), or selectable timed interrupt routine (file 5) because a fault will occur.

Suspend (SUS)



Execution Times (µsec) when:

True	False
10.85	7.87

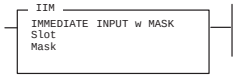
When this instruction is executed, it causes the controller to enter the Suspend Idle mode and stores the Suspend ID in word 7 (S:7) at the status file. All outputs are de-energized.

Use this instruction to trap and identify specific conditions for program debugging and system troubleshooting.

Entering Parameters

Enter a suspend ID number from –32,768 to +32,767 when you program the instruction.

Immediate Input with Mask (IIM)



Execution Times (µsec) when:

True	False
35.72	6.78

This instruction allows you to update data prior to the normal input scan. Data from a specified input is transferred through a mask to the input data file, making the data available to instructions following the IIM instruction in the ladder program.

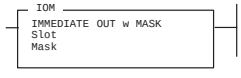
For the mask, a 1 in an input's bit position passes data from the source to the destination. A 0 inhibits data from passing from the source to the destination.

Entering Parameters

For all micro controllers specify I1:0.0. For 16 I/O controllers, I1:0/0–9 are valid and I1:0/10–15 are considered unused inputs. (They do not physically exist.) For 32 I/O controllers, I1:0/0–15 and I1:1/0–3 are valid. Specify I1:1 if you want to immediately update the last four input bits.

Mask – Specify a Hex constant or register address.

Immediate Output with Mask (IOM)



Execution Times (µsec) when:

True	False
41.59	6.78

This instruction allows you to update the outputs prior to the normal output scan. Data from the output image is transferred through a mask to the specified outputs. The program scan then resumes.

Entering Parameters

For all micro controllers specify O0:0.0. For 16 I/O controllers, O0:0/0–5 are valid and O0:0/6–15 are considered unused outputs. (They do not physically exist.) For 32 I/O controllers, O0:0/0–11 are valid and O0:0/12–15 are considered unused outputs.

Mask – Specify a Hex constant or register address.

Program Flow Control Instructions in the Paper Drilling Machine Application Example

This section provides ladder rungs to demonstrate the use of program flow control instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will be adding to the main program in file 2. The new rungs are needed to call the other subroutines containing the logic necessary to run the machine.

```
Rung 2:5
Calls the drill sequence subroutine.  This subroutine manages the
operation of a drilling sequence and restarts the conveyor upon
completion of the drilling sequence

|                                                  +JSR---------------+ |
|--------------------------------------------------+JUMP TO SUBROUTINE+-|
|                                                  |SBR file number  6| |
|                                                  +------------------+ |
Rung 2:6
Calls the subroutine that tracks the amount of wear on the current
drill bit.

|                                                  +JSR---------------+ |
|--------------------------------------------------+JUMP TO SUBROUTINE+-|
|                                                  |SBR file number  7| |
|                                                  +------------------+ |
Rung 2:7

|                                                                       |
|-------------------------------------+END+----------------------------|
|                                                                       |
```

11 Using Application Specific Instructions

This chapter contains general information about the application specific instructions and explains how they function in your application program. Each of the instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the application specific instructions in use.

Application Specific Instructions

Instruction		Purpose	Page
Mnemonic	**Name**		
BSL and BSR	Bit Shift Left and Bit Shift Right	Loads a bit of data into a bit array, shifts the pattern of data through the array, and unloads the last bit of data in the array. The BSL shifts data to the left and the BSR shifts data to the right.	**11–5**
SQO and SQC	Sequencer Output and Sequencer Compare	Control sequential machine operations by transferring 16-bit data through a mask to image addresses.	**11–7**
SQL	Sequencer Load	Capture referenced conditions by manually stepping the machine through its operating sequences.	**11–13**
STD and STE	Selectable Timer Interrupt Disable and Enable	Output instructions, associated with the Selectable Timed Interrupt function. STD and STE are used to prevent an STI from occurring during a portion of the program.	**11–18**

Continued on the next page.

Instruction		Purpose	Page
Mnemonic	Name		
STS	Selectable Timer Interrupt Start	Initiates a Selectable Timed Interrupt.	**11–20**
INT	Interrupt Subroutine	Associated with Selectable Timed Interrupts or HSC Interrupts.	**11–20**

About the Application Specific Instructions

These instructions simplify your ladder program by allowing you to use a single instruction or pair of instructions to perform common complex operations.

In this chapter you will find a general overview preceding groups of instructions. Before you learn about the instructions in each of these groups, we suggest that you read the overview. This chapter contains the following overviews:

- Bit Shift Instructions Overview
- Sequencer Instructions Overview
- Selectable Timed Interrupt (STI) Function Overview

Bit Shift Instructions Overview

The following general information applies to bit shift instructions.

Entering Parameters

Enter the following parameters when programming these instructions:

- **File** is the address of the bit array you want to manipulate. You must use the file indicator (#) in the bit array address.
- **Control** is the address of the control element that stores the status byte of the instruction, the size of the array (in number of bits). Note that the control address should not be used for any other instruction.

The control element is shown below.

	15 13	11 10	00
Word 0	EN DN	ER UL	Not used
Word 1	Size of bit array (number of bits)		
Word 2	Reserved		

Status bits of the control element may be addressed by mnemonic. They include:

- **Unload Bit UL** (bit 10) is the instruction's output.
- **Error Bit ER** (bit 11), when set, indicates the instruction detected an error such as entering a negative number for the length or position. Avoid using the unload bit when this bit is set.
- **Done Bit DN** (bit 13), when set, indicates the bit array shifted one position.
- **Enable Bit EN** (bit 15) is set on a false-to-true transition of the rung and indicates the instruction is enabled.

When the register shifts and input conditions go false, the enable, done, and error bits are reset.

- **Bit Address** is the address of the source bit. The status of this bit is inserted in either the first (lowest) bit position (BSL) or last (highest) bit position (BSR).
- **Length** (size of bit array) is the number of bits in the bit array, up to 1680 bits. A length value of 0 causes the input bit to be transferred to the UL bit.

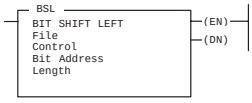
  A length value that points past the end of the programmed file causes a major error to occur. *If you alter a length value with your ladder program, make certain that the altered value is valid.*

  The instruction invalidates all bits beyond the last bit in the array (as defined by the length) up to the next word boundary.

Effects on Index Register S:24

The shift operation clears the index register S:24 to zero.

Bit Shift Left (BSL)



When the rung goes from false-to-true, the controller sets the enable bit (EN bit 15) and the data block is shifted to the left (to a higher bit number) one bit position. The specified bit at the bit address is shifted into the first bit position. The last bit is shifted out of the array and stored in the unload bit (UL bit 10). The shift is completed immediately.

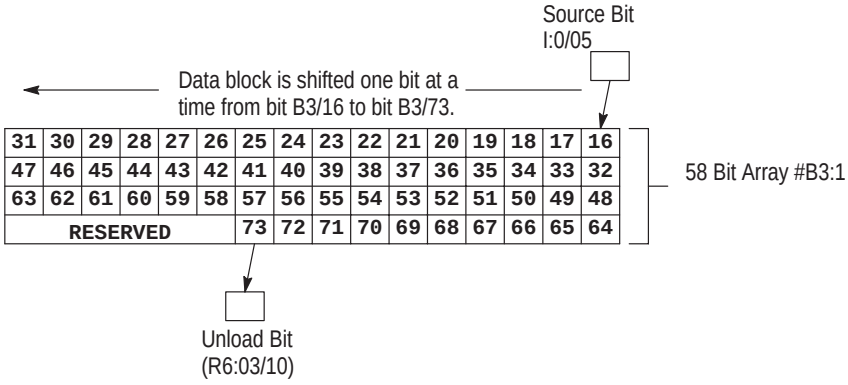
For wraparound operation, set the bit address to the last bit of the array or to the UL bit.

Execution Times (µsec) when:

True	False
53.71+ 5.24/word	19.80

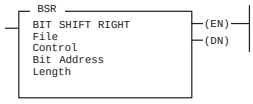
Operation

The following figure shows the operation of the BSL instruction shown above.



If you wish to shift more than one bit per scan, you must create a loop in your application using the JMP, LBL, and CTU instructions.

Bit Shift Right (BSR)



Execution Times (µsec) when:

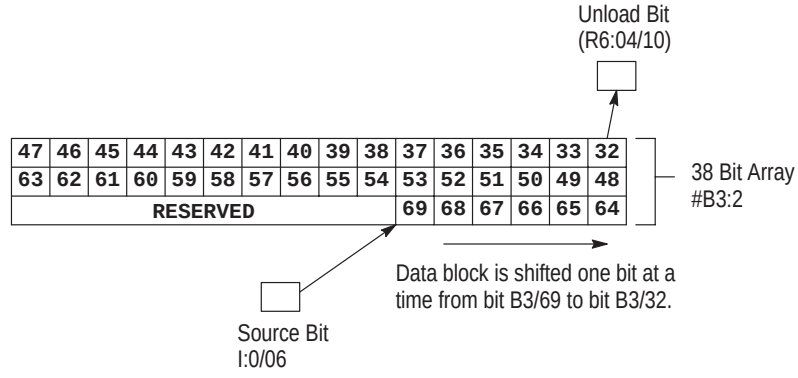
True	False
53.34+ 3.98/word	19.80

When the rung goes from false-to-true, the controller sets the enable bit (EN bit 15) and the data block is shifted to the right (to a lower bit number) one bit position. The specified bit at the bit address is shifted into the last bit position. The first bit is shifted out of the array and stored in the unload bit (UL bit 10). The shift is completed immediately.

For wraparound operation, set the bit address to the first bit of the array or to the UL bit.

Operation

The following figure shows the operation of the BSR instruction shown above.



If you wish to shift more than one bit per scan, you must create a loop in your application using the JMP, LBL, and CTU instructions.

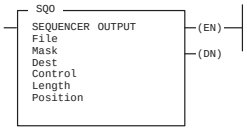
Sequencer Instructions Overview

The following general information applies to sequencer instructions.

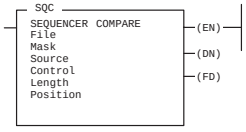
Effects on Index Register S:24

The value present in the index register S:24 is overwritten when the sequencer instruction is true. The index register value will equal the position value of the instruction.

Sequencer Output (SQO) and Sequencer Compare (SQC)



These instructions transfer 16-bit data to word addresses for the control of sequential machine operations.



Execution Times (µsec) when:

	True	False
SQO	60.52	27.40
SQC	60.52	27.40

Entering Parameters

Enter the following parameters when programming these instructions:

- **File** is the address of the sequencer file. You must use the file indicator (#) for this address.

  Sequencer file data is used as follows:

Instruction	Sequencer File Stores
SQO	Data for controlling outputs
SQC	Reference data for monitoring inputs

- **Mask** (SQO, SQC) is a hexadecimal code or the address of the mask word or file through which the instruction moves data. Set mask bits to pass data and clear mask bits to prevent the instruction from operating on corresponding destination bits. Use a mask word or file if you want to change the mask according to application requirements.

  If the mask is a file, its length will be equal to the length of the sequencer file. The two files track automatically.

- **Source** is the address of the input word or file for a SQC from which the instruction obtains data for comparison to its sequencer file.
- **Destination** is the address of the output word or file for a SQO to which the instruction moves data from its sequencer file.

Note

You can address the mask, source, or destination of a sequencer instruction as a word or file. If you address it as a file (using file indicator #), the instruction automatically steps through the source, mask, or destination file.

- **Control** (SQO, SQC) is the control structure that stores the status byte of the instruction, the length of the sequencer file, and the current position in the file. You should not use the control address for any other instruction.

	15 13 11 08 00
Word 0	EN DN ER FD
Word 1	Length of sequencer file
Word 2	Position

Status bits of the control structure include:

- **Found Bit FD** (bit 08) – SQC only. When the status of all non-masked bits in the source address match those of the corresponding reference word, the FD bit is set. This bit is assessed each time the SQC instruction is evaluated while the rung is true.
- **Error Bit ER** (bit 11) is set when the controller detects a negative position value, or a negative or zero length value. When the ER bit is set, the minor error bit (S5:2) is also set. Both bits must be cleared.
- **Done Bit DN** (bit 13) is set by the SQO or SQC instruction after it has operated on the last word in the sequencer file. It is reset on the next false-to-true rung transition after the rung goes false.
- **Enable EN** (bit 15) is set by a false-to-true rung transition and indicates the SQO or SQC instruction is enabled.

- **Length** is the number of steps of the sequencer file starting at position 1. The maximum number you can enter is 104 words. Position 0 is the startup position. The instruction resets (wraps) to position 1 at each cycle completion.
- **Position** is the word location or step in the sequencer file from/to which the instruction moves data.

You may use the RES instruction to reset a sequencer. All control bits (except FD) will be reset to zero. The Position will also be set to zero. Program the address of your control register in the RES (e.g., R6:0).

Using SQO

This output instruction steps through the sequencer file whose bits have been set to control various output devices.

When the rung goes from false-to-true, the instruction increments to the next step (word) in the sequencer file. Data stored there is transferred through a mask to the destination address specified in the instruction. Data is written to the destination word every time the instruction is executed.

The done bit is set when the last word of the sequencer file is transferred. On the next false-to-true rung transition, the instruction resets the position to step one.

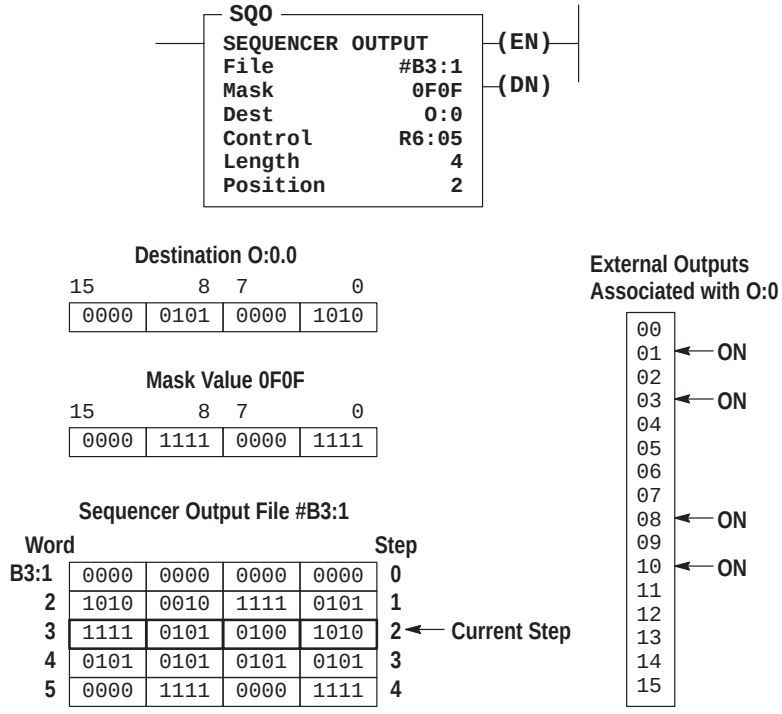
If the position is equal to zero at startup, when you switch the controller from the program mode to the run mode instruction operation depends on whether the rung is true or false on the first scan.

- If true, the instruction transfers the value in step zero.
- If false, the instruction waits for the first rung transition from false-to-true and transfers the value in step one.

The bits mask data when reset and pass data when set. The instruction will not change the value in the destination word unless you set mask bits.

The mask can be fixed or variable. It will be fixed if you enter a hexadecimal code. It will be variable if you enter an element address or a file address for changing the mask with each step.

The following figure indicates how the SQO instruction works.



Using SQC

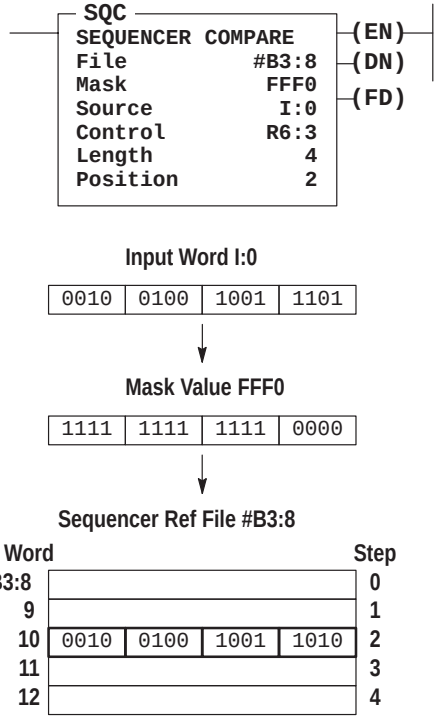
When the status of all non-masked bits in the source word match those of the corresponding reference word, the instruction sets the found bit (FD) in the control word. Otherwise, the found bit (FD) is cleared.

The bits mask data when reset and pass data when set.

The mask can be fixed or variable. If you enter a hexadecimal code, it is fixed. If you enter an element address or a file address for changing the mask with each step, it is variable.

When the rung goes from false-to-true, the instruction increments to the next step (word) in the sequencer file. Data stored there is transferred through a mask and compared against the source for equality. While the rung remains true, the source is compared against the reference data for every scan. If equal, the FD bit is set in the SQCs control counter.

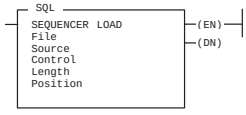
Applications of the SQC instruction include machine diagnostics. The following figure explains how the SQC instruction works.



The SQC FD bit is set when the instruction detects that an input word matches (thru mask) its corresponding reference word.

The FD bit R6:3/FD is set in this example, since the input word matches the sequencer reference value using the mask value.

Sequencer Load (SQL)



The SQL instruction stores 16-bit data into a sequencer load file at each step of sequencer operation. The source of this data can be an I/O or internal word address, a file address, or a constant.

Execution Times (µsec) when:

True	False
53.41	28.12

Entering Parameters

Enter the following parameters when programming this instruction:

- **File** is the address of the sequencer file. You must use the file indicator (#) for this address.
- **Source** can be a word address, file address, or a constant (–32768 to 32767).

  If the source is a file address, the file length equals the length of the sequencer load file. The two files step automatically, according to the position value.
- **Length** is the number of steps of the sequencer load file (and also of the source if the source is a file address), starting at position 1. The maximum number you can enter is 104 words. Position 0 is the startup position. The instruction resets (wraps) to position 1 at each cycle completion.
- **Position** is the word location or step in the sequencer file to which data is moved.
- **Control** is a control file address. The status bits, length value, and position value are stored in this element. Do not use the control file address for any other instruction.

The control element is shown below:

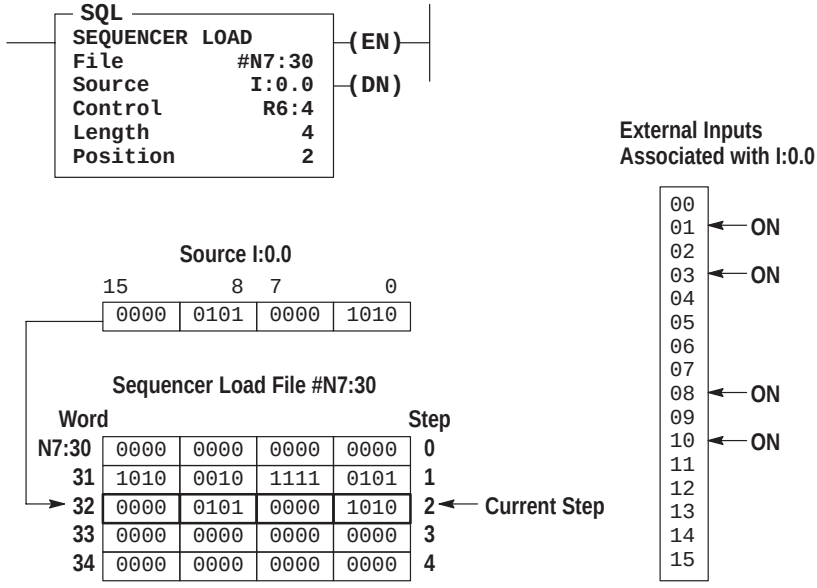
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Word 0	EN			DN			ER									
Word 1	Length															
Word 2	Position															

Status bits of the control structure include:

- **Error Bit ER** (bit 11) is set when the controller detects a negative position value, or a negative or zero length value. When the ER bit is set, the minor error bit (S5:2) is also set. Both bits must be cleared.
- **Done Bit DN** (bit 13) is set after the instruction has operated on the last word in the sequencer load file. It is reset on the next false-to-true rung transition after the rung goes false.
- **Enable Bit EN** (bit 15) is set on a false-to-true transition of the SQL rung and reset on a true-to-false transition.

Operation

Instruction parameters have been programmed in the SQL instruction shown below. Input word I:0.0 is the source. Data in this word is loaded into integer file #N7:30 by the sequencer load instruction.



When rung conditions change from false-to-true, the SQL enable bit (EN) is set. The control element R6:4 increments to the next position in the sequencer file, and loads the contents of source I:0.0 into the corresponding location in the file. The SQL instruction continues to load the current data into this location each scan that the rung remains true. When the rung goes false, the enable bit (EN) is reset.

The instruction loads data into a new file element at each false-to-true transition of the rung. When step 4 is completed, the done bit (DN) is set. Operation cycles to position 1 at the next false-to-true transition of the rung after position 4.

If the source were a file address such as #N7:40, files #N7:40 and #N7:30 would both have a length of 5 (0–4) and would track through the steps together per the position value.

Selectable Timed Interrupt (STI) Function Overview

The Selectable Timed Interrupt (STI) function allows you to interrupt the scan of the application program automatically, on a periodic basis, to scan a subroutine file. Afterwards the controller resumes executing the application program from the point where it was interrupted.

Basic Programming Procedure for the STI Function

To use the STI function in your application file:

1. Enter the desired ladder rungs in File 5. (File 5 is designated for the STI subroutine.)

2. Enter the setpoint (the time between successive interrupts) in word S:30 of the status file. The range is 10–2550 ms (entered in10 ms increments). A setpoint of zero disables the STI function.

Note *The setpoint value must be a longer time than the execution time of the STI subroutine file, or a minor error bit is set.*

Operation

After you restore your program and enter the REM Run or REM Test mode, the STI begins operation as follows:

1. The STI timer begins timing.

2. When the STI interval expires, the program scan is interrupted and the STI subroutine file is scanned; the STI timer is reset.

3. If while executing the STI (file 5), another STI interrupt occurs the STI pending bit (S:2/0) is set.

4. If while an STI is pending, the STI timer expires, the STI lost bit (S:5/10) is set.

5. When the STI subroutine scan is completed, scanning of the program resumes at the point where it left off, unless an STI is pending. In this case the subroutine is immediately scanned again.

6. The cycle repeats.

For identification of your STI subroutine, include an INT instruction as the first instruction on the first rung of the file.

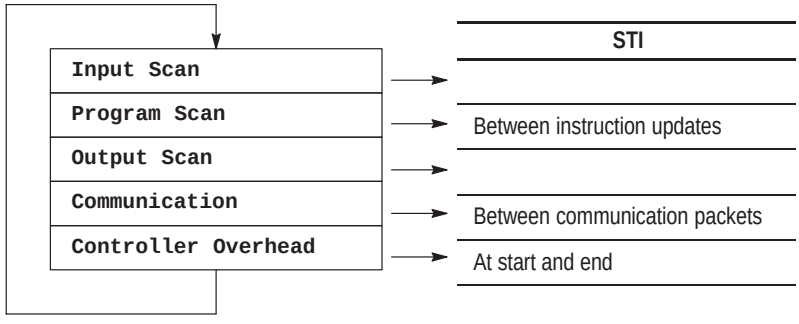
STI Subroutine Content

The STI subroutine contains the rungs of your application logic. You can program any instruction inside the STI subroutine except a TND instruction. IIM or IOM instructions are needed in an STI subroutine if your application requires immediate update of input or output points. End the STI subroutine with an RET instruction.

JSR stack depth is limited to 3. You may call other subroutines to a level 3 deep from an STI subroutine.

Interrupt Latency and Interrupt Occurrences

Interrupt latency is the interval between the STI timeout and the start of the interrupt subroutine. STI interrupts can occur at any point in your program, but not necessarily at the same point on successive interrupts. The table below shows the interaction between an interrupt and the controller operating cycle.



Events in the processor operating cycle

Note that STI execution time adds directly to the overall scan time. During the latency period, the controller is performing operations that cannot be disturbed by the STI interrupt function.

Interrupt Priorities

Interrupt priorities are as follows:

1. User Fault Routine
2. High-Speed Counter
3. Selectable Timed Interrupt

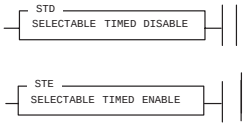
An executing interrupt can only be interrupted by an interrupt having a higher priority.

Status File Data Saved

Data in the following words is saved on entry to the STI subroutine and re-written upon exiting the STI subroutine.

- S:0 Arithmetic flags
- S:13 and S:14 Math register
- S:24 Index register

Selectable Timed Disable (STD) and Enable (STE)



These instructions are generally used in pairs. The purpose is to create zones in which STI interrupts *cannot* occur.

Execution Times (µsec) when:

	True	False
STD	6.69	3.16
STE	10.13	3.16

Using STD

When true, this instruction resets the STI enable bit and prevents the STI subroutine from executing. When the rung goes false, the STI enable bit remains reset until a true STS or STE instruction is executed. The STI timer continues to operate while the enable bit is reset.

Using STE

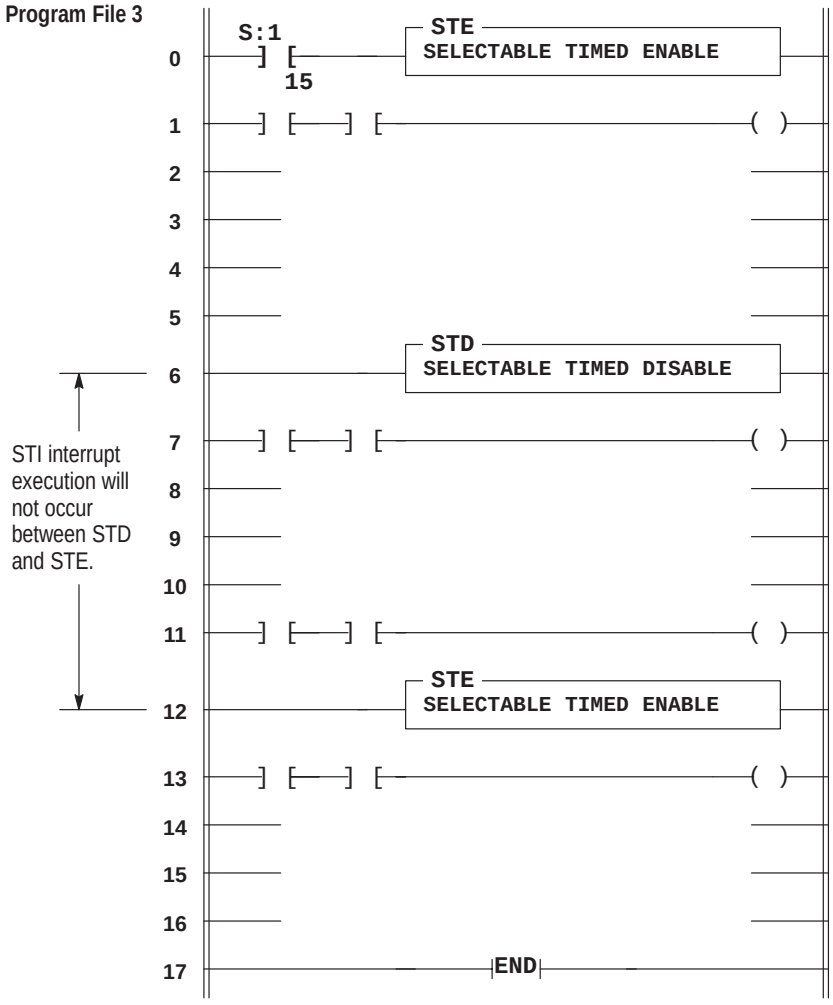
This instruction sets the STI enable bit and allows execution of the STI subroutine. When the rung goes false, the STI enable bit remains set until a true STD instruction is executed. This instruction has no effect on the operation of the STI timer or setpoint. When the enable bit is set, the first execution of the STI subroutine can occur at any point up to the full STI interval.

STD/STE Zone Example

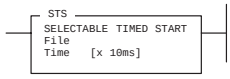
In the program that follows, the STI function is in effect. The STD and STE instructions in rungs 6 and 12 are included in the ladder program to avoid having STI subroutine execution at any point in rungs 7 through 11.

The STD instruction (rung 6) resets the STI enable bit and the STE instruction (rung 12) sets the enable bit again. The STI timer increments and may time out in the STD zone, setting the pending bit S:2/0 and lost bit S:5/10.

The first pass bit S:1/15 and the STE instruction in rung 0 are included to insure that the STI function is initialized following a power cycle. You should include this rung any time your program contains an STD/STE zone or an STD instruction.


Program File 3
S:1
15
STE
SELECTABLE TIMED ENABLE
STD
SELECTABLE TIMED DISABLE
STE
SELECTABLE TIMED ENABLE
END
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
STI interrupt execution will not occur between STD and STE.

Selectable Timed Start (STS)



Execution Times (µsec) when:

True	False
24.59	6.78

Use the STS instruction to condition the start of the STI timer upon entering the REM Run mode – rather than starting automatically. You can also use it to set up or change setpoint/frequency of the STI routine that will be executed when the STI timer expires.

This instruction is not required to configure a basic STI interrupt application.

The STS instruction requires you to enter the parameter for the STI setpoint. Upon a true execution of the rung, this instruction enters the setpoint in the status file (S:30), overwriting the existing data. At the same time, the STI timer is reset and begins timing; at timeout, the STI subroutine execution occurs. When the rung goes false, the STI function remains enabled at the setpoint you've entered in the STS instruction.

Interrupt Subroutine (INT)

Execution Times (µsec) when:

True	False
1.45	0.99

This instruction serves as a label or identifier of a program file as an interrupt subroutine (INT label) versus a regular subroutine (SBR label).

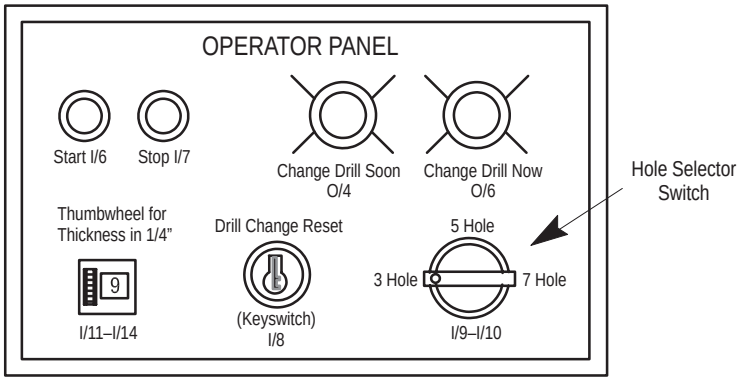
This instruction has no control bits and is always evaluated as true. The instruction must be programmed as the first instruction of the first rung of the subroutine. Use of this instruction is optional; however, we recommend using it.

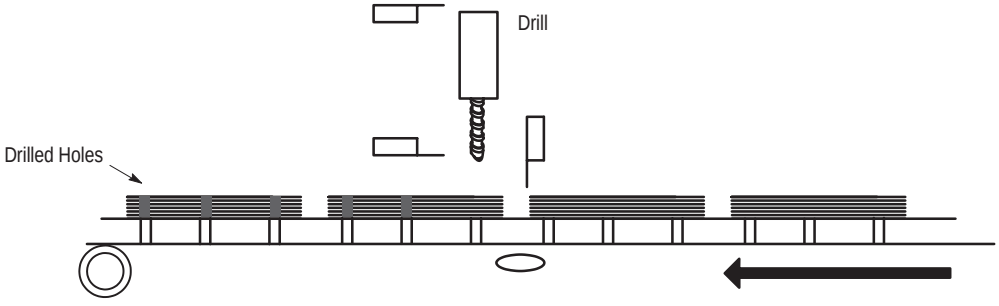
Application Specific Instructions in the Paper Drilling Machine Application Example

This section provides ladder rungs to demonstrate the use of application specific instructions. The rungs are part of the paper drilling machine application example described in appendix E. You will begin a subroutine in file 4.

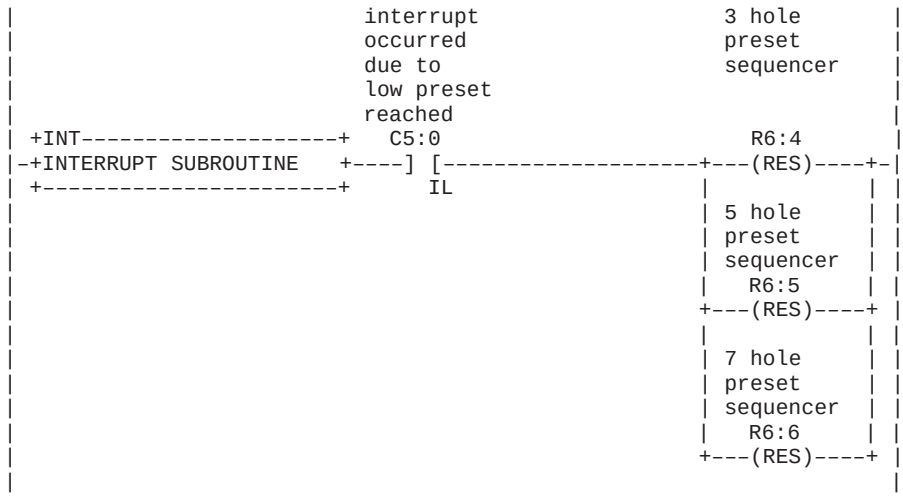
This portion of the subroutine tells the conveyor where to stop to allow a hole to be drilled. The stop positions will be different for each hole pattern (3 hole, 5 hole, 7 hole), so separate sequencers are used to store and access each of the three hole patterns.

Note *Address I:0/10 is only valid for 32 I/O controllers. If you use a 16 I/O controller, only the 5 hole drill pattern can be used.*

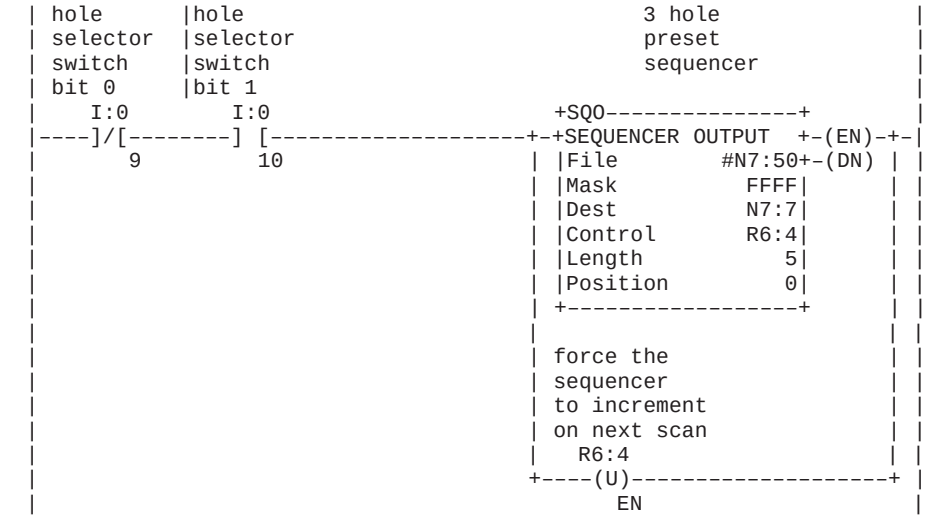




Rung 4:0
Resets the hole count sequencers each time that the low preset is reached. The low preset has been set to zero to cause an interrupt to occur each time that a reset occurs. The low preset is reached anytime that a reset C5:0 or hardware reset occurs. This ensures that the first preset value is loaded into the HSC at each entry into the REM Run mode and each time that the external reset signal is activated.

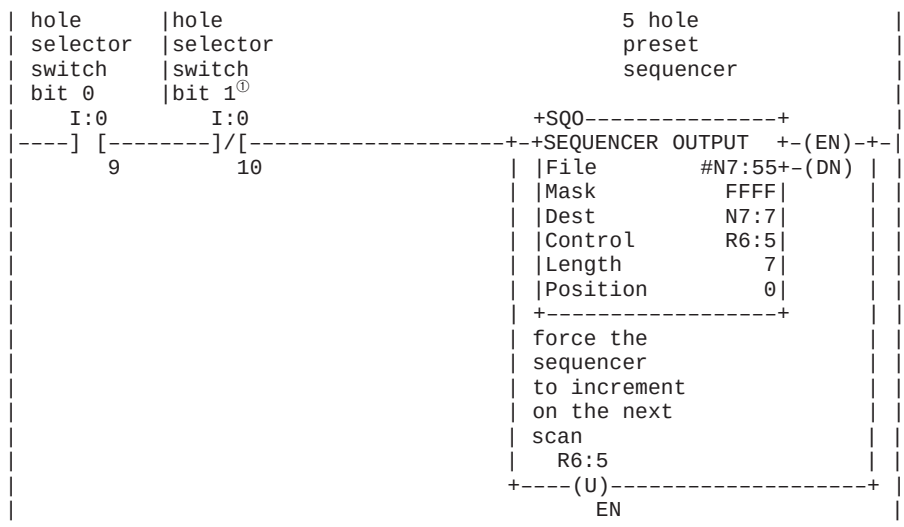


Rung 4:1[1]
Keeps track of the hole number that is being drilled and loads the correct HSC preset based on the hole count. This rung is only active when the ″hole selector switch″ is in the ″3-hole″ position. The sequencer uses step 0 as a null step upon reset. It uses the last step as a ″go forever″ in anticipation of the ″end of manual″ hard wired external reset.



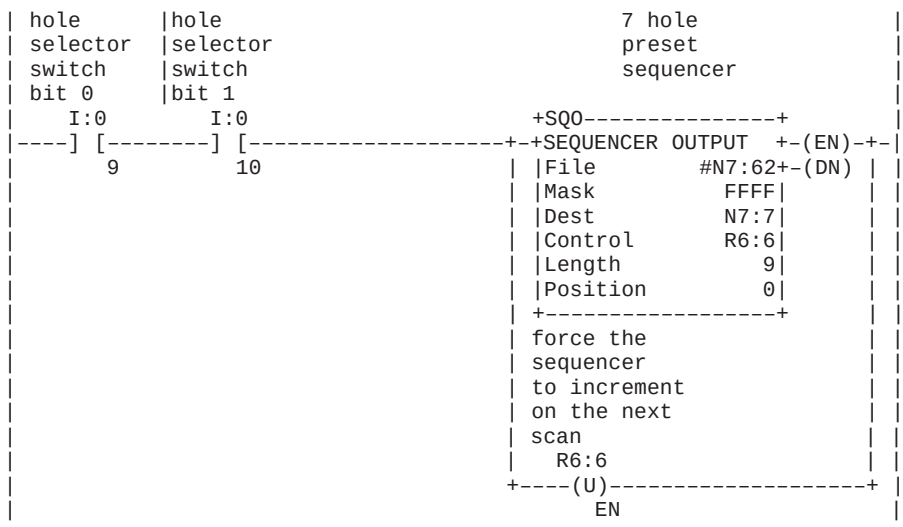
[1] This rung accesses I/O only available with 32 I/O controllers. Do not include this rung if you are using a 16 I/O controller.

Rung 4:2
Is identical to the previous rung except that it is only active when the ”hole selector switch” is in the ”5-hole” position.



① This instruction accesses I/O only available with 32 I/O controllers. Do not include this instruction if you are using a 16 I/O controller.

Rung 4:3[1][2]
Is identical to the 2 previous rungs except that it is only active when the ”hole selector switch” is in the ”7-hole” position.



[1] This rung accesses I/O only available with 32 I/O controllers. Do not include this rung if you are using a 16 I/O controller.

[2] More rungs will be added to this subroutine at the end of chapter 12.

Notes:

12 Using High-Speed Counter Instructions

This chapter contains general information about the high-speed counter instructions and explains how they function in your application program. Each of the instructions includes information on:

- what the instruction symbol looks like
- typical execution time for the instruction
- how to use the instruction

In addition, the last section contains an application example for a paper drilling machine that shows the high-speed counter instructions in use.

High-Speed Counter Instructions

Instruction		Purpose	Page
Mnemonic	Name		
HSC	High-Speed Counter	Applies configuration to the high-speed counter hardware, updates the image accumulator, enables counting when the HSC is true, and disables counting when the HSC rung is false.	**12–6**
HSL	High-Speed Counter Load	Configures the low and high presets, the output patterns, and mask bit patterns.	**12–18**
RES	High-Speed Counter Reset	Writes a zero to the hardware accumulator and image accumulator.	**12–21**
RAC	High-Speed Counter Reset Accumulator	Writes the value specified to the hardware accumulator and image accumulator.	**12–22**
HSE **HSD**	High-Speed Counter Interrupt Enable High-Speed Counter Interrupt Disable	Enables or disables execution of the high-speed counter interrupt subroutine when a high preset, low preset, overflow, or underflow is reached.	**12–23**
OTE	Update High-Speed Counter Image Accumulator	Provides you with real-time access to the hardware accumulator value by updating the image accumulator.	**12–24**

About the High-Speed Counter Instructions

The high-speed counter instructions used in your ladder program configure, control, and monitor the controllers' hardware counter. The hardware counter's accumulator increments or decrements in response to external input signals. When the high-speed counter is enabled, data table counter C5:0 is used by the ladder program for monitoring the high-speed counter accumulator and status. The high-speed counter operates *independent* of the controller scan.

When using the high-speed counter, make sure you adjust your input filters accordingly. See page A–7 for more information on input filters.

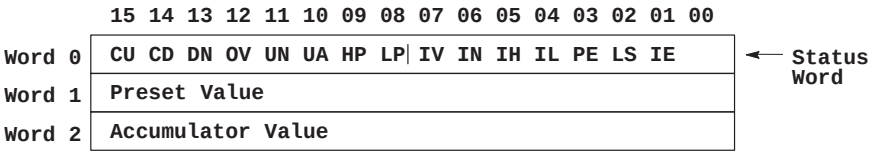
Before you learn about these instructions, read the overview that follows on the next page. Refer to page 2–24 for information on wiring your controller for high-speed counter applications.

High-Speed Counter Instructions Overview

Use the high-speed counter to detect and store narrow (fast) pulses, and its specialized instructions to initiate other control operations based on counts reaching preset values. These control operations include the automatic and immediate execution of the high-speed counter interrupt routine (file 4) and the immediate update of outputs based on a source and mask pattern you set.

Counter Data File Elements

The high-speed counter instructions reference counter C5:0. The HSC instruction is fixed at C5:0. It is comprised of three words. Word 0 is the status word, containing 15 status bits. Word 1 is the preset value. Word 2 is the accumulated value. Once assigned to the HSC instruction, C5:0 is not available as an address for any other counter instructions.


15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Word 0
CU CD DN OV UN UA HP LP IV IN IH IL PE LS IE
Status Word
Word 1
Preset Value
Word 2
Accumulator Value

CU = Counter Up Enable Bit
CD = Counter Down Enable Bit
DN = High Preset Reached Bit
OV = Overflow Occurred Bit
UN = Underflow Occurred Bit
UA = Update High-Speed Counter Accumulator Bit
HP = Accumulator ≥ High Preset Bit
LP = Accumulator ≤ Low Preset Bit
IV = Overflow Caused High-Speed Counter Interrupt Bit
IN = Underflow Caused High-Speed Counter Interrupt Bit
IH = High Preset Reached Caused Interrupt Bit
IL = Low Preset Reached Caused Interrupt Bit
PE = High-Speed Counter Interrupt Pending Bit
LS = High-Speed Counter Interrupt Lost Bit
IE = High-Speed Counter Interrupt Enable Bit

Counter preset and accumulated values are stored as signed integers.

Using Status Bits

The high-speed counter status bits are retentive. When the high-speed counter is first configured, bits 3–7, 14, and 15 are reset and bit 1 (IE) is set.

- **Counter Up Enable Bit CU** (bit 15) is used with all of the high-speed counter types. If the HSC instruction is true, the CU bit is set to one. If the HSC instruction is false, the CU bit is set to zero. Do not write to this bit.
- **Counter Down Enable Bit CD** (bit 14) is used with the Bidirectional Counters (modes 3–8). If the HSC instruction is true, the CD bit is set to one. If the HSC instruction is false, the CD bit is set to zero. Do not write to this bit.
- **High Preset Reached Bit DN** (bit 13) For the Up Counters (modes 1 and 2), this bit is an edge triggered latch bit. This bit is set when the high preset is reached. You can reset this bit with an OTU instruction or by executing an RAC or RES instruction.

  The DN bit is a reserved bit for all other Counter options (modes 3–8).

- **Overflow Occurred Bit OV** (bit 12) For the Up Counters (modes 1 and 2), this bit is set by the controller when the high preset is reached if the DN bit is set.

Tip

For the Bidirectional Counters (modes 3–8), the OV bit is set by the controller after the hardware accumulator transitions from 32,767 to –32,768. You can reset this bit with an OTU instruction or by executing an RAC or RES instruction for both the up and bidirectional counters.

- **Underflow Occurred Bit UN** (bit 11) is a reserved bit for the Up Counters (modes 1 and 2). Do not write to this bit.

Tip

For the Bidirectional Counters (modes 3–8), the UN bit is set by the controller when the hardware accumulator transitions from –32,768 to +32,767. You can reset this bit with an OTU instruction or by executing an RAC or RES instruction.

- **Update High-Speed Counter Accumulator Bit UA** (bit 10) is used with an OTE instruction to update the instruction image accumulator value with the hardware accumulator value. (The HSC instruction also performs this operation each time the rung with the HSC instruction is evaluated as true.)
- **Accumulator ≥ High Preset Bit HP** (bit 9) is a reserved bit for all Up Counters(modes 1 and 2).

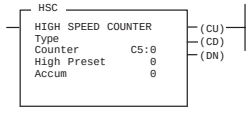
  For the Bidirectional Counters (modes 3–8), if the hardware accumulator becomes greater than or equal to the high preset, the HP bit is set. If the hardware accumulator becomes less than the high preset, the HP bit is reset by the controller. Do not write to this bit. (Exception – you can set or reset this bit during the initial configuration of the HSC instruction. See page 12–6 for more information.)

- **Accumulator ≤ Low Preset Bit LP** (bit 8) is a reserved bit for all Up Counters.

  For the Bidirectional Counters, if the hardware accumulator becomes less than or equal to the low preset, the LP bit is set by the controller. If the hardware accumulator becomes greater than the low preset, the LP bit is reset by the controller. Do not write to this bit. (Exception – you can set or reset this bit during the initial configuration of the HSC instruction. See page 12–6 for more information.)

- **Overflow Caused High-Speed Counter Interrupt Bit IV** (bit 7) is set to identify an overflow as the cause for the execution of the high-speed counter interrupt routine. The IN, IH, and IL bits are reset by the controller when the IV bit is set. Examine this bit at the start of the high-speed counter interrupt routine (file 4) to determine why the interrupt occurred.
- **Underflow caused User Interrupt Bit IN** (bit 6) is set to identify an underflow as the cause for the execution of the high-speed counter interrupt routine. The IV, IH, and IL bits are reset by the controller when the IN bit is set. Examine this bit at the start of the high-speed counter interrupt routine (file 4) to determine why the interrupt occurred.
- **High Preset Reached Caused User Interrupt Bit IH** (bit 5) is set to identify a high preset reached as the cause for the execution of the high-speed counter interrupt routine. The IV, IN, and IL bits are reset by the controller when the IH bit is set. Examine this bit at the start of the high-speed counter interrupt routine (file 4) to determine why the interrupt occurred.
- **Low Preset Reached Caused High-Speed Counter Interrupt Bit IL** (bit 4) is set to identify a low preset reached as the cause for the execution of the high-speed counter interrupt routine. The IV, IN, and IH bits are reset by the controller when the IL bit is set. Examine this bit at the start of the high-speed counter interrupt routine (file 4) to determine why the interrupt occurred.
- **High-Speed Counter Interrupt Pending Bit PE** (bit 3) is set to indicate that a high-speed counter interrupt is waiting for execution. This bit is cleared by the controller when the high-speed counter interrupt routine begins executing. This bit is reset if an RAC or RES instruction is executed. Do not write to this bit.
- **High-Speed Counter Interrupt Lost Bit LS** (bit 2) is set if a high-speed counter interrupt occurs while the PE bit is set. You can reset this bit with an OTU instruction or by executing an RAC or RES instruction.
- **High-Speed Counter Interrupt Enable Bit IE** (bit 1) is set when the high-speed counter interrupt is enabled to run when a high-speed counter interrupt condition occurs. It is reset when the interrupt is disabled. This bit is also set when the high-speed counter is first configured. Do not write to this bit.

High-Speed Counter (HSC)



Execution Times (µsec) when:

True	False
21.00	21.00

Use this instruction to configure the high-speed counter. Only one HSC instruction can be used in a program. The high-speed counter is not operational until the first true execution of the HSC instruction. When the HSC rung is false, the high-speed counter is disabled from *counting*, but all other HSC features are operational.

The Counter address of the HSC instruction is fixed at C5:0.

After the HSC is configured, the image accumulator (C5:0.ACC) is updated with the current hardware accumulator value every time the HSC instruction is evaluated as true or false.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Type** indicates the counter selected. Refer to page 12–7 for making your high-speed counter selection. Each type is available with reset and hold functionality.
- **High Preset** is the accumulated value that triggers a user-specified action such as updating outputs or generating a high-speed counter interrupt.
- **Accumulator** is the number of accumulated counts.

The following terminology is used in the following table to indicate the status of counting:

- Up↑ – increments by 1 when the input energizes (edge).
- Down↑ – decrements by 1 when the input energizes (edge).
- Reset↑ – resets the accumulator to zero when the input energizes (edge).
- Hold – disables the high-speed counter from counting while the input is energized (level).
- Count – increments or decrements by 1 when the input energizes (edge).
- Direction – allows up counts when the input is de-energized and down counts while the input is energized (level).
- A – input pulse in an incremental (quadrature) encoder (edge/level).
- B – input pulse in an incremental (quadrature) encoder (edge/level).
- Z – reset pulse in an incremental (quadrature) encoder (edge/level).
- ↑ – the signal is active on the rising edge only (off to on).

The table below lists the function key you press to choose the type of high-speed counter you want.

High-Speed Counter Type and Function Key	High-Speed Counter Functionality	Input Terminal Used			
		I/0	I/1	I/2	I/3
[F1] Up	Up Counter operation uses a single-ended input.	Up↑	Not Used	Not Used	Not Used
[F2] Up (with reset and hold)	Up Counter operation uses a single input with external reset and hold inputs.	Up↑	Not Used	Reset↑	Hold
[F3] Pulse and direction	Bidirectional operation uses both pulse and direction inputs.	Count↑	Direction	Not Used	Not Used
[F4] Pulse and direction (with external reset and hold)	Bidirectional operation uses both pulse and direction inputs with external reset and hold inputs.	Count↑	Direction	Reset↑	Hold
[F5] Up and down	Bidirectional operation uses both up and down direction inputs.	Up↑	Down↑	Not Used	Not Used
[F6] Up and down (with external reset and hold)	Bidirectional operation uses both up and down pulse inputs with external reset and hold inputs.	Up↑	Down↑	Reset↑	Hold
[F7] Encoder	Bidirectional operation uses quadrature encoder inputs.	A	B	Not Used	Not Used
[F8] Encoder (with external reset and hold)	Bidirectional operation uses both quadrature encoder inputs with external reset and hold inputs.	A	B	Z	Hold

One difference between Up Counters and Bidirectional Counters is that for Bidirectional Counters the accumulator and preset values are not changed by the high-speed counter when the presets are reached. The RAC and HSL instructions must be used for this function. The Up Counters clear the accumulator and re-load the high preset values whenever the preset is reached.

Using the Up Counter and the Up Counter with Reset and Hold

Up counters are used when the parameter being measured is uni-directional, such as material being fed into a machine or as a tachometer recording the number of pulses over a given time period.

Both types of Up Counters operate identically, except that the Up Counter with reset and hold uses external inputs 2 and 3.

For the Up Counter, each Off-to-On state change of input I:0/0 adds 1 to the accumulator until the high preset is reached. The accumulator is then automatically reset to zero. The Up Counter operates in the 0 to +32,767 range inclusive and can be reset to zero using the Reset (RES) instruction.

When the HSC instruction is first executed true, the:

- Accumulator C5:0:0.ACC is loaded to the hardware accumulator.
- High preset C5:0:0.PRE is loaded to the hardware high preset.

Operation

If you move data to the high preset without using the RAC instruction (with a MOV) after the high-speed counter has been configured, the data is loaded to the instruction image but is *not* loaded to the hardware. The modified high preset value is not loaded to the hardware until the existing hardware high preset is reached, or an RAC or RES instruction is executed.

The high preset value loaded to the hardware must be between 1 and 32,767 inclusive or an error `INVALID PRESETs LOADED TO HIGH SPEED COUNTER` (37H) occurs. Any value between –32,768 and +32,767 inclusive can be loaded to the hardware accumulator.

The Following Condition	Occurs when
A high preset is reached	either the hardware accumulator transitions from the hardware high preset –1 to the hardware high preset, or
	the hardware accumulator is loaded with a value greater than or equal to the hardware high preset, or
	the hardware high preset is loaded with a value that is less than or equal to the hardware accumulator.

When a high preset is reached, no counts are lost.

- Hardware and instruction accumulators are reset.
- Instruction high preset is loaded to the hardware high preset.
- If the DN bit is not set, the DN bit is set. The IH bit is also set and the IL, IV, and IN bits are reset.
- If the DN bit is already set, the OV bit is set. The IV bit is also set and the IL, IV and IN bits are reset.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled.

The following tables summarize what the input state must be for the corresponding high-speed counter action to occur:

Up Counter

Input State					High-Speed Counter Action
Input Count (I/O)	Input Direction (I/1)	Input Reset (I/2)	Input Hold (I/3)	HSC Rung	
Turning Off-to-On	NA	NA	NA	True	Count Up
NA	NA	NA	NA	False	Hold Count
Off, On, or Turning Off	NA	Off, On, or Turning Off	NA	NA	Hold Count

NA (Not Applicable)

Up Counter with Reset and Hold

Input state					High-Speed Counter Action
Input Count (I/O)	Input Direction (I/1)	Input Reset (I/2)	Input Hold (I/3)	HSC Rung	
Turning Off-to-On	NA	Off, On, or Turning Off	Off	True	Count Up
NA	NA	Off, On, or Turning Off	On	NA	Hold Count
NA	NA	Off, On, or Turning Off	NA	False	Hold Count
Off, On, or Turning Off	NA	Off, On, or Turning Off	NA	NA	Hold Count
NA	NA	Turning On	NA	NA	Reset to 0

NA (Not Applicable)

Using the Bidirectional Counter and the Bidirectional Counter with Reset and Hold

Bidirectional counters are used when the parameter being measured can either increment or decrement. For example, a package entering and leaving a storage bin is counted to regulate flow through the area.

The Bidirectional Counters operate identically except for the operation of inputs 1 and 0. For the Pulse and Direction type, input 0 provides the pulse and input 1 provides the direction. For the Up and Down type, input 0 provides the Up count and input 1 provides the Down count. Both types are available with and without reset and hold. Refer to page 12–7 for more information regarding Bidirectional Counter types.

For the Bidirectional Counters, both high and low presets are used. The low preset value must be less than the high preset value or an error `INVALID PRESETs LOADED TO HIGH SPEED COUNTER` (37H) occurs.

Bidirectional Counters operate in the –32,768 to +32,767 range inclusive and can be reset to zero using the Reset (RES) instruction.

Operation

When the HSC instruction is first executed true, the:

- Instruction accumulator is loaded to the hardware accumulator.
- Instruction high preset is loaded to the hardware high preset.

After the first true HSC instruction execution, data can only be transferred to the hardware accumulator via an RES or RAC instruction, or to the hardware high and low presets via the HSL instruction.

Any instruction accumulator value between –32,768 and +32,767 inclusive can be loaded to the hardware.

The Following Condition	Occurs when
A high preset is reached	either the hardware accumulator transitions from the hardware high preset –1 to the hardware high preset, or
	the hardware accumulator is loaded with a value greater than or equal to the hardware high preset, or
	the hardware high preset is loaded with a value that is less than or equal to the hardware accumulator.

When a high preset is reached, the:

- HP bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IH bit is set and the IL, IV, and IN bits are reset.

Unlike the Up Counters, the accumulator value does not get reset and the high preset value does not get loaded from the image to the hardware high preset register.

The Following Condition	Occurs when
A low preset is reached	either the hardware accumulator transitions from the hardware low preset +1 to the hardware low preset, or
	the hardware accumulator is loaded with a value less than or equal to the hardware low preset, or
	the hardware low preset is loaded with a value that is greater than or equal to the hardware accumulator.

When the low preset is reached, the:

- LP bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IL bit is set and the IH, IV, and IN bits are reset.

An overflow occurs when the hardware accumulator transitions from +32,767 to –32,768. When an overflow occurs, the:

- OV bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IV bit is set and the IH, IL, and IN bits are reset.

An underflow occurs when the hardware accumulator transitions from –32,768 to +32,767. When an underflow occurs, the:

- UN bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IN bit is set and the IH, IL, and IV bits are reset.

The following tables summarize what the input state must be for the corresponding high-speed counter action to occur:

Bidirectional Counter (Pulse/direction)

Input State					High-Speed Counter Action
Input Count (I/0)	Input Direction (I/1)	Input Reset (I/2)	Input Hold (I/3)	HSC Rung	
Turning Off-to-On	Off	NA	NA	True	Count Up
Turning Off-to-On	On	NA	NA	True	Count Down
NA	NA	NA	NA	False	Hold Count
Off, On, or Turning Off	NA	NA	NA	NA	Hold Count

NA (Not Applicable)

Bidirectional Counter with Reset and Hold (Pulse/direction)

Input State					High-Speed Counter Action
Input Count (I/0)	Input Direction (I/1)	Input Reset (I/2)	Input Hold (I/3)	HSC Rung	
Turning Off-to-On	Off	Off, On, or Turning Off	Off	True	Count Up
Turning Off-to-On	On	Off, On, or Turning Off	Off	True	Count Down
NA	NA	Off, On, or Turning Off	NA	False	Hold Count
NA	NA	Off, On, or Turning Off	On	NA	Hold Count
Off, On, or Turning Off	NA	Off, On, or Turning Off	NA	NA	Hold Count
NA	NA	Turning On	NA	NA	Reset to 0

NA (Not Applicable)

Bidirectional Counter (Up/down count)

Input State			High-Speed Counter Action
Input Up Count (I/0)	Input Down Count (I/1)	HSC Rung	
Turning Off-to-On	Off, On, or Turning Off	True	Count Up
Off, On, or Turning Off	Turning Off-to-On	True	Count Down
NA	NA	False	Hold Count
Off, On, or Turning Off	Off, On, or Turning Off	NA	Hold Count

NA (Not Applicable)

Bidirectional Counter with Reset and Hold (Up/down count)

Input State					High-Speed Counter Action
Input Up Count (I/0)	Input Down Count (I/1)	Input Reset (I/2)	Input Hold (I/3)	HSC Rung	
Turning Off-to-On	Off, On, or Turning Off	Off, On, or Turning Off	Off	True	Count Up
Off, On, or Turning Off	Turning Off-to-On	Off, On, or Turning Off	Off	True	Count Down
NA	NA	Off, On, or Turning Off	NA	False	Hold Count
NA	NA	Off, On, or Turning Off	On	NA	Hold Count
Off, On, or Turning Off	Off, On, or Turning Off	Off, On, or Turning Off	NA	NA	Hold Count
NA	NA	Turning On	NA	NA	Reset to 0

NA (Not Applicable)

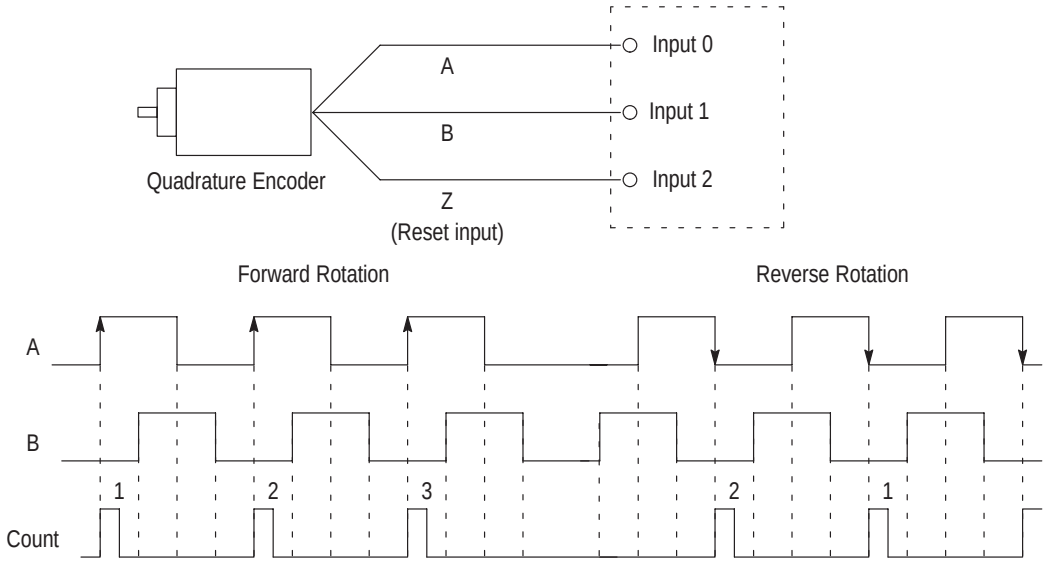
When up and down input pulses occur simultaneously, the high-speed counter counts up, then down.

Using the Bidirectional Counter with Reset and Hold with a Quadrature Encoder

The Quadrature Encoder is used for determining direction of rotation and position for rotating, such as a lathe. The Bidirectional Counter counts the rotation of the Quadrature Encoder.

Bidirectional Counters operate in the –32,768 to +32,767 range inclusive and can be reset to zero using the reset (RES) instruction. The following figure shows a quadrature encoder connected to inputs 0, 1, and 2. The count direction is determined by the phase angle between A and B. If A leads B, the counter increments. If B leads A, the counter decrements.

The counter can be reset using the Z input. The Z outputs from the encoders typically provide one pulse per revolution.



Operation

For the Bidirectional Counters, both high and low presets are used. The low preset value must be less than the high preset value or an error `INVALID PRESETs LOADED TO HIGH SPEED COUNTER` (37H) occurs.

When the HSC instruction is first executed true, the:

- Instruction accumulator is loaded to the hardware accumulator.
- Instruction high preset is loaded to the hardware high preset.

Any instruction accumulator value between –32,768 and +32,767 inclusive can be loaded to the hardware.

After the first true HSC instruction execution, data can only be transferred to the hardware accumulator via an RES or RAC instruction, or to the hardware high and low presets via the HSL instruction.

The Following Condition	Occurs when
A high preset is reached	either the hardware accumulator transitions from the hardware high preset –1 to the hardware high preset, or
	the hardware accumulator is loaded with a value greater than or equal to the hardware high preset, or
	the hardware high preset is loaded with a value that is less than or equal to the hardware accumulator.

When a high preset is reached, the:

- HP bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IH bit is set and the IL, IN, and IV bits are reset.

Unlike the Up Counters, the accumulator value does not reset and the high preset value does not get loaded from the image to the hardware high preset register.

The Following Condition	Occurs when
A low preset is reached	either the hardware accumulator transitions from the hardware low preset +1 to the hardware low preset, or
	the hardware accumulator is loaded with a value less than or equal to the hardware low preset, or
	the hardware low preset is loaded with a value that is greater than or equal to the hardware accumulator.

When a low preset is reached, the:

- LP bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IL bit is set and the IH, IN, and IV bits are reset.

An overflow occurs when the hardware accumulator transitions from +32,767 to –32,768. When an overflow occurs, the:

- OV bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IV bit is set and the IH, IL, and IN bits are reset.

An underflow occurs when the hardware accumulator transitions from –32,768 to +32,767. When an underflow occurs, the:

- UN bit is set.
- High-speed counter interrupt file (program file 4) is executed if the interrupt is enabled. The IN bit is set and the IH, IL, and IV bits are reset.

The following tables summarize what the input state must be for the corresponding high-speed counter action to occur:

Bidirectional Counter (Encoder)

Input State			High-Speed Counter Action
Input A (I/0)	Input B (I/1)	HSC Rung	
Turning On	Off	True	Count Up
Turning Off	Off	True	Count Down
NA	On	NA	Hold Count
NA	NA	False	Hold Count
Off or On	NA	NA	Hold Count

NA (Not Applicable)

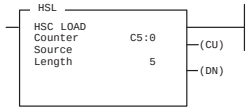
Bidirectional Counter with Reset and Hold (Encoder)

Input State					High-Speed Counter Action
Input A (I/0)	Input B (I/1)	Input Z (I/2)	Input Hold (I/3)	HSC Rung	
Turning On	Off	Off	Off	True	Count Up
Turning Off	Off	Off	Off	True	Count Down
Off or On	NA	Off	NA	NA	Hold Count
NA	On	Off	NA	NA	Hold Count
NA	NA	Off	NA	False	Hold Count
NA	NA	Off	On	NA	Hold Count
Off	Off	On①	NA	NA	Reset to 0

NA (Not Applicable)

① The optional hardware high-speed counter reset is the logical coincidence of $\overline{A} \times \overline{B} \times Z$.

High-Speed Counter Load (HSL)



This instruction allows you to set the low and high presets, low and high output source, and the output mask. When either a high or low preset is reached, you can instantly update selected outputs.

Execution Times (µsec) when:

True	False
66.00	7.00

If you are using the HSL instruction with the Up Counter, the high preset must be ≥ 1 and ≤ +32,767 or an error `INVALID PRESETs LOADED TO HIGH SPEED COUNTER` (37H) occurs. For the bidirectional counters, the high preset must be greater than the low preset or an error `INVALID PRESETs LOADED TO HIGH SPEED COUNTER` (37H) occurs.

The Counter referenced by this instruction has the same address as the HSC instruction counter and is fixed at C5:0.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** is an address that identifies the first of five data words used by the HSL. The source can be either an integer or binary file element.
- **Length** is the number of elements starting from the source. This number is always 5.

Operation

The HSL instruction allows you to *configure* the high-speed counter to instantaneously and automatically update external outputs whenever a high or low preset is reached. The physical outputs are automatically updated in less than 30 µs. (The physical turn-on time of the outputs is *not* included in this amount.) The output image is then automatically updated at the next poll for user interrupts or IOM instruction, whichever occurs first.

With this instruction, you can change the high preset for the up counters or both the high and low presets for Bidirectional Counters during run. You can also modify the output mask configuration during run.

The source address is either an integer or binary file element. For example, if N7:5 is selected as the source address, the additional parameters for the execution of this instruction would appear as shown in the following table.

Parameter Image Location	Up Counter Only	Bidirectional Counters	Description
N7:5	Output Mask	Output Mask	Identifies which group of four bits in the output file (word 0) are controlled. 000F=bits 3–0 00F0=bits 7–4 0003=bits 0 and 1 00FF= bits 7–0
N7:6	Output Source	Output High Source	(Up count.) The status of bits in this word are written "through" the mask to the actual outputs.
N7:7	High Preset	High Preset	(Up count.) When the accumulator reaches this value, the output source are written through the output mask to the actual outputs, and the HSC subroutine (file 4) will be scanned.
N7:8	Reserved	Output Low Source	(Down count.) The status of bits in this word are written "through" the mask to the actual outputs.
N7:9	Reserved	Low Preset	(Down count.) When the accumulator reaches this value, the output source are written through the output mask to the actual outputs, and the HSC subroutine (file 4) will be scanned.

The bits in the output mask directly correspond to the physical outputs. If a bit is set to 1, the corresponding output can be changed by the high-speed counter. If a bit is set to 0, the corresponding output cannot be changed by the high-speed counter.

The bits in the high and low sources also directly correspond to the physical outputs. The high source is applied when the high preset is reached. The low source is applied when the low preset is reached. The final output states are determined by applying the output source over the mask and updating only the unmasked outputs (those with a 1 in the mask bit pattern).

You can always change the state of the outputs via the user program or programming device regardless of the output mask. The high-speed counter only modifies selected outputs and output image bits based on source and mask bit patterns when the presets are reached. The last device that changes the output image (i.e., user program or high-speed counter) determines the actual output pattern.

Forces override any output control from either the high-speed counter or from the output image. Forces may also be applied to the high-speed counter inputs. Forced inputs are recognized by the high-speed counter (e.g., a forced count input off and on increments the high-speed accumulator).

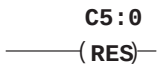
The high-speed counter hardware is updated immediately when the HSL instruction is executed regardless of high-speed counter type (Up Counter or Bidirectional Counter). For the Up Counters, the last two registers are ignored since the low preset does not apply.

If a fault occurs due to the HSL instruction, the HSL parameters are not loaded to the high-speed counter hardware. You can use more than one HSL instruction in your program. The HSL instructions can have different image locations for the additional parameters.

Do not change a preset value *and* an output mask/source with the same HSL instruction as the accumulator is approaching the old preset value.

If the high-speed counter is enabled and the HSL instruction is evaluated true, the high-speed counter parameters in the HSL instruction are applied immediately without stopping the operation of the high-speed counter. If the same HSL instruction is being used to change the high-speed counter controlled mask/source and the preset, the mask/source is changed first and the preset second. (The preset is changed within 40μs after the mask/source.) If the original preset is reached after the new mask/source is applied but before the new preset is applied, the new outputs are applied immediately.

High-Speed Counter Reset (RES)



Execution Times (µsec) when:

True	False
51.00	6.00

The RES instruction allows you to write a zero to the hardware accumulator and image accumulator.

The Counter referenced by this instruction has the same address as the HSC instruction counter and is entered as C5:0.

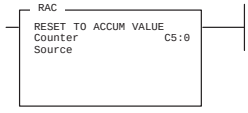
Operation

Execution of this instruction immediately:

- removes pending high-speed counter interrupts
- resets the hardware and instruction accumulators
- reset the PE, LS, OV, UN, and DN status bits
- loads the instruction high preset to the hardware high preset (if the high-speed counter is configured as an up counter)
- resets the IL, IH, IN, or IV status bits

You can have more than one RES instruction in your program.

High-Speed Counter Reset Accumulator (RAC)



This instruction allows you to write a specific value to the hardware accumulator and image accumulator.

The Counter referenced by this instruction has the same address as the HSC instruction counter and is fixed at C5:0.

Execution Times (µsec) when:

True	False
56.00	6.00

Entering Parameters

Enter the following parameter when programming this instruction:

- **Source** represents the value that is loaded to the accumulator. The source can be a constant or an address.

Operation

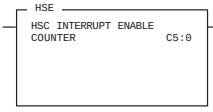
Execution of the RAC:

- removes pending high-speed counter interrupts
- resets the PE, LS, OV, UN, and DN status bits
- loads a new accumulator value to the hardware and instruction image
- loads the instruction high preset to the hardware high preset (if the high-speed counter is configured as an Up Counter)
- resets the IL, IH, IN, or IV status bits

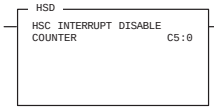
The source can be a constant or any integer element in files 0–7. The hardware and instruction accumulators are updated with the new accumulator value immediately upon instruction execution.

You can have more than one RAC instruction per program referencing the same source or different sources.

High-Speed Counter Interrupt Enable (HSE) and Disable (HSD)



These instructions enable or disable a high-speed counter interrupt when a high preset, low preset, overflow, or underflow is reached. Use the HSD and HSE in pairs to provide accurate execution for your application.



The Counter referenced by these instructions have the same address as the HSC instruction counter and is fixed at C5:0.

Execution Times (µsec) when:

	True	False
HSE	10.00	7.00
HSD	8.00	7.00

Using HSE

Operation

When the high-speed counter interrupt is enabled, user subroutine (program file 4) is executed when:

- A high or low preset is reached.
- An overflow or underflow occurs.

When in Test Single Scan mode and in an idle condition, the high-speed counter interrupt is held off until the next scan trigger is received from the programming device. The high-speed counter accumulator counts while idle.

If the HSE is subsequently executed after the pending bit is set, the interrupt is executed immediately.

The default state of the high-speed counter interrupt is enabled (the IE bit is set to 1).

If the high-speed counter interrupt routine is executing and another high-speed counter interrupt occurs, the second high-speed counter interrupt is saved but is considered pending. (The PE bit is set.) The second interrupt is executed immediately after the first one is finished executing. If a high-speed counter interrupt occurs while a high-speed counter interrupt is pending, the most recent high-speed counter interrupt is lost and the LS bit is set.

Using HSD

Operation

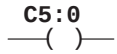
The HSD instruction disables the high-speed counter interrupt, preventing the interrupt subroutine from being executed.

If the HSE is subsequently executed after the pending bit is set, the interrupt is executed immediately.

This HSD instruction does not cancel an interrupt, but results in the pending bit (C5:0/3) being set when:

- A high or low preset is reached.
- An overflow or underflow occurs.

Update High-Speed Counter Image Accumulator (OTE)



Execution Times (µsec) when:

True	False
12.00	7.00

When an OUT bit instruction is addressed for the high-speed counter (C5:0) UA bit, the value in the hardware accumulator is written to the value in the image accumulator (C5:0.ACC). This provides you with real-time access to the hardware accumulator value. This is in addition to the automatic transfer from the hardware accumulator to the image accumulator that occurs each time the HSC instruction is evaluated.

Operation

This instruction transfers the hardware accumulator to the instruction accumulator. When the OTE/UA instruction is executed true, the hardware accumulator is loaded to the instruction image accumulator (C5:0.ACC).

What Happens to the HSC When Going to REM Run Mode

Once initialized, the HSC instruction retains its previous state when going through a mode change or power cycle. This means that the HSC Accumulator (C5:0.ACC) and High Preset values are retained. Outputs under the direct control of the HSC also retain their previous state. The Low Preset Reached and High Preset Reached bits (C0/LP and C0/HP) are also retained. They are examined by the HSC instruction during the high-speed counter's first true evaluation in the REM Run mode to differentiate a retentive REM Run mode entry from an external or initial Accumulator (C5:0.ACC) modification.

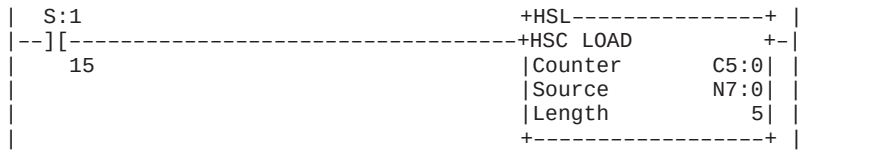
At the first true HSL instruction execution after going-to-run, the Low Preset is initialized to –32,768 and the output mask and high and low output patterns are initialized to zero. Use the HSL instruction during the first pass to restore any values necessary for your application.

You can modify the behavior of the high-speed counter at REM Run mode entry by adjusting the HSC parameters prior to the first true execution of the HSC instruction. The following example ladder rungs demonstrate different ways to adjust the HSC parameters.

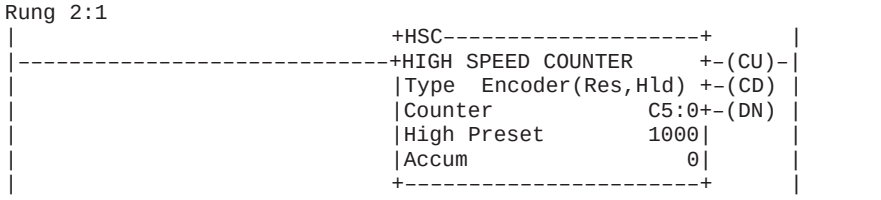
Example 1

To enter the REM Run mode and have the HSC Outputs, ACC, and Interrupt Subroutine resume their previous state, apply the following:

```
(Rung 2:0)
No action required.  (Remember that all OUT instructions are zeroed
when entering the REM Run mode. Use SET/RST instructions in place of
OUT instructions in your conditional logic requiring retention.)
```



Rung 2:1



Example 2

To enter the REM Run mode and retain the HSC ACC value while having the HSC Outputs and Interrupt Subroutine reassert themselves, apply the following:

```
Rung 2:0
Unlatch the C5:0/HP and C5:0/LP bits during the first scan BEFORE the
HSC instruction is executed for the first time.

|  S:1                                          +HSL---------------+  |
|--][-------------------------------------------+HSC LOAD          +- |
|    15                                         |Counter       C5:0|  |
|                                               |Source        N7:0|  |
|                                               |Length           5|  |
|                                               +------------------+  |

Rung 2:1
|  S:1                                                      C5:0     |
|--][-----------------------------------------------------+-(U)--+|--|
|    15                                                   |    HP |  |
|                                                         | C5:0  |  |
|                                                         +--(U)--+  |
|                                                              LP    |

Rung 2:2
|                                      +HSC---------------------+      |
|--------------------------------------+HIGH SPEED COUNTER     +-(CU)-|
|                                      |Type  Encoder (Res,Hld)+-(CD) |
|                                      |Counter            C5:0+-(DN) |
|                                      |High Preset        1000|      |
|                                      |Accum                 0|      |
|                                      +----------------------+      |
```

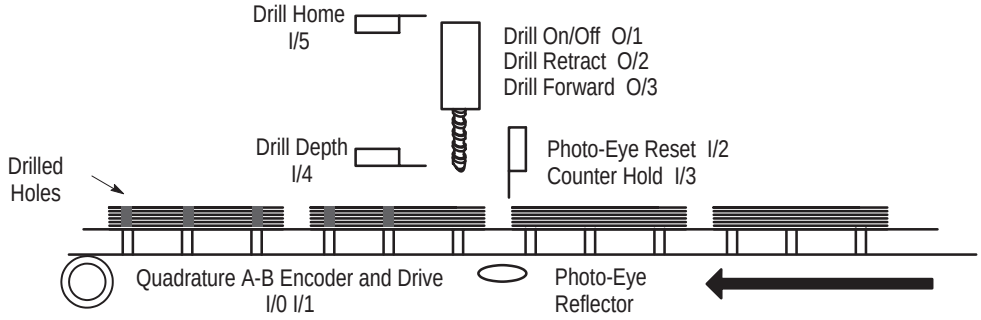
Example 3

To enter the REM Run mode and have the HSC ACC and Interrupt Subroutine resume their previous state, while externally initializing the HSC outputs, apply the following:

```
Rung 2:0
Unlatch or Latch the output bits under HSC control during the first
scan after the HSC instruction is executed for the first time. (Note,
you could place this rung before the HSC instruction; however, this is
not recommended.)

|  S:1                                          +HSL---------------+ |
|--][-------------------------------------------+HSC LOAD          +-|
|    15                                         |Counter       C5:0| |
|                                               |Source        N7:0| |
|                                               |Length           5| |
|                                               +------------------+ |

Rung 2:1
|                                    +HSC--------------------+      |
|------------------------------------+HIGH SPEED COUNTER     +-(CU)-|
|                                    |Type  Encoder (Res,Hld)+-(CD) |
|                                    |Counter            C5:0+-(DN) |
|                                    |High Preset        1000|      |
|                                    |Accum                 0|      |
|                                    +-----------------------+      |

Rung 2:2
This rung is programmed with the knowledge of an HSL mask of 0007
(Outputs
0–2 are used) and initializes the HSC outputs each REM Run mode entry.
Outputs O/0 and O/1 are off, while Output O/2 is on.

|  S:1                                                      O:0      |
|--][-----------------------------------------------------+--(U)--+|--|
|    15                                                   |    0   |  |
|                                                         |  O:0   |  |
|                                                         +--(U)---+  |
|                                                         |    1   |  |
|                                                         |  O:0   |  |
|                                                         +--(L)---+  |
|                                                               2     |
```

High-Speed Counter Instructions in the Paper Drilling Machine Application Example

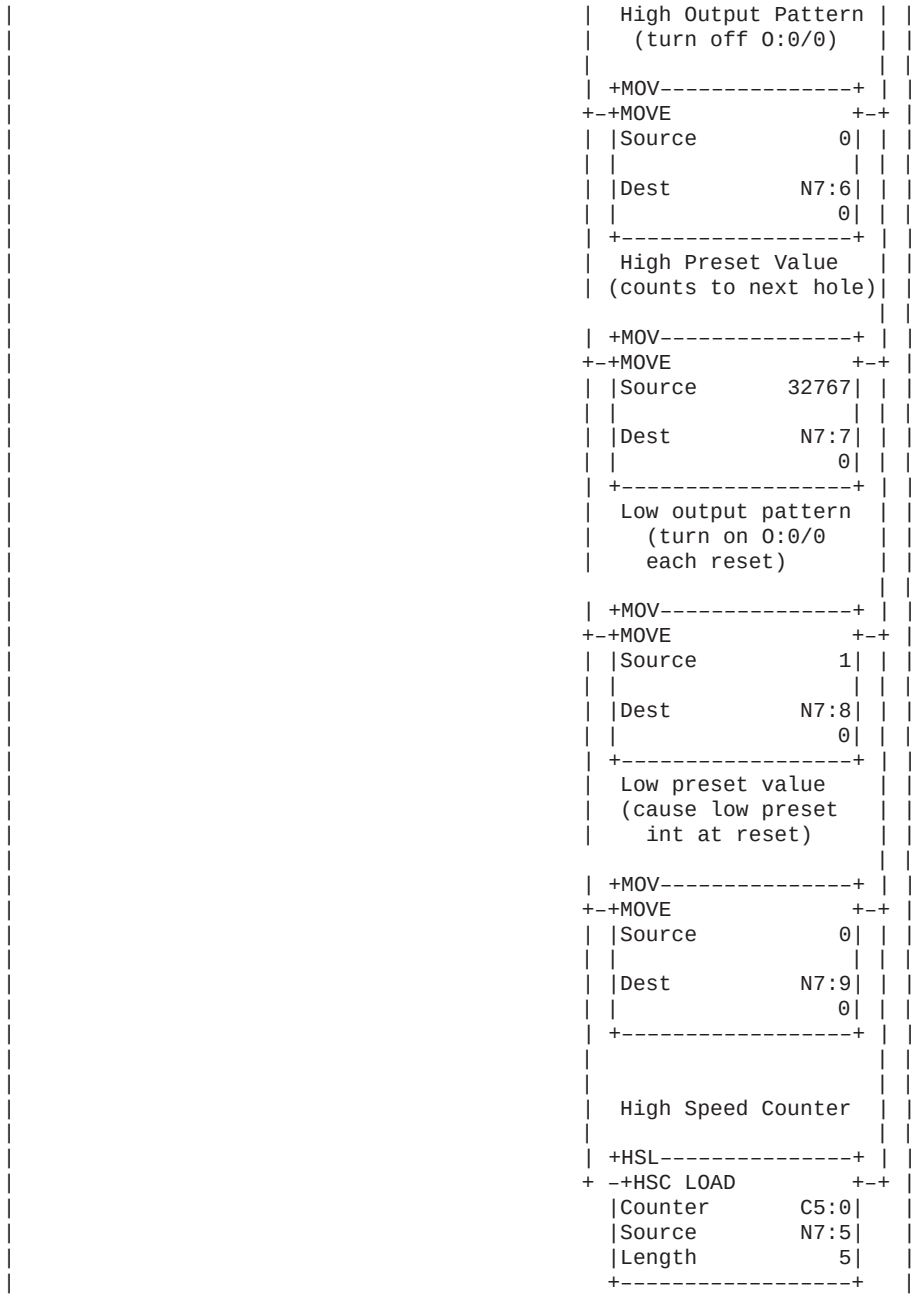
The ladder rungs in this section demonstrate the use of the HSC instruction in the paper drilling machine application example started in chapter 6. Refer to appendix E for the complete paper drilling machine application example.



20226

The main program file (file 2) initializes the HSC instruction, monitors the machine start and stop buttons, and calls other subroutines necessary to run the machine. Refer to the comments preceding each rung for additional information.

```
Rung 2:0
Initializes the high-speed counter each time the REM Run mode is
entered.  The high-speed counter data area (N7:5 – N7:9) corresponds
with the starting address (source address) of the HSL instruction.  The
HSC instruction is disabled each entry into the REM run mode until the
first time that it is executed as true.  (The high preset was ”pegged”
on initialization to prevent a high preset interrupt from occurring
during the initialization process.)

| 1’st                                              Output Mask        |
| Pass                                              (only use bit 0    |
|                                                    ie. O:0/0)        |
|    S:1                                         +MOV---------------+   |
|----] [-----------------------------------------+-+MOVE              +-+-|
|      15                                        | |Source           1| | |
|                                                | |                  | | |
|                                                | |Dest          N7:5| | |
|                                                | |                 0| | |
|                                                | +------------------+ | |
```


High Output Pattern
(turn off O:0/0)
MOV
MOVE
Source 0
Dest N7:6
0
High Preset Value
(counts to next hole)
MOV
MOVE
Source 32767
Dest N7:7
0
Low output pattern
(turn on O:0/0
each reset)
MOV
MOVE
Source 1
Dest N7:8
0
Low preset value
(cause low preset
int at reset)
MOV
MOVE
Source 0
Dest N7:9
0
High Speed Counter
HSL
HSC LOAD
Counter C5:0
Source N7:5
Length 5

Rungs 2.0 and 2.2 are required to write several parameters to the high-speed counter data file area. These two rungs are conditioned by the first pass bit during one scan when the controller is going from REM program to REM Run mode.

```
Rung 2:1
This HSC instruction is not placed in the high-speed counter interrupt
subroutine.  If this instruction were placed in the interrupt
subroutine, the high-speed counter could never be started or
initialized (because an interrupt must first occur in order to scan the
high-speed counter interrupt subroutine).

|                                          High Speed Counter        |
|                                       +HSC--------------------+    |
|---------------------------------------+HIGH SPEED COUNTER     +-(CU)-|
|                                       |Type  Encoder (Res,Hld)+-(CD) |
|                                       |Counter            C5:0+-(DN) |
|                                       |High Preset        1250|    |
|                                       |Accum                 1|    |
|                                       +-----------------------+    |
Rung 2:2
Forces a high-speed counter low preset interrupt to occur each REM Run
mode entry. An interrupt can only occur on the transition of the
high-speed counter accum to a preset value (accum reset to 1, then 0).
This is done to allow the high-speed counter interrupt subroutine
sequencers to initialize.  The order of high-speed counter
initialization is:  (1)load high-speed counter parameters  (2)execute
HSL instruction  (3)execute true HSC instruction  (4)(optional) force
high-speed counter interrupt to occur.

| 1'st                                      High Speed Counter       |
| Pass                                                               |
|    S:1                                   +RAC------------------+   |
|----] [-----------------------------------+-+RESET TO ACCUM VALUE +-+-|
|      15                                  | |Counter          C5:0| | |
|                                          | |Source              1| | |
|                                          | |                     | | |
|                                          | +---------------------+ | |
|                                          |    High Speed           | |
|                                          |    Counter              | |
|                                          |   C5:0                  | |
|                                          +---(RES)-----------------+ |
```

The high-speed counter is used to control the conveyer position. The high-speed counter counts pulses supplied by the conveyer's encoder via hardware inputs I:0/0 and I:0/1. Hardware inputs I:0/2 (reset) and I:0/3 (hold) are connected to a photo-switch ensuring the HSC instruction only counts encoder pulses when a manual is in front of the drill and that the high-speed counter is reset at the leading edge of each manual.

The high-speed counter clears the conveyer drive output bit (O:0/0) each time a high preset is reached. As a result, the drive decelerates and stops the conveyer motor. The high-speed counter clears the output within microseconds ensuring accuracy and repeatability.

The high-speed counter sets the conveyer drive output bit (O:0/0) each time a low preset is reached. As a result, the drive accelerates and maintains the conveyer motor.

When the manual has travelled the specified distance set by the high-speed counter high preset value, the high-speed counter interrupt subroutine signals the main program to perform the drilling sequence. For more information regarding the interrupt subroutine used in this program, refer to the application example in chapter 11.

This example uses the Quadrature Encoder with reset and hold instruction. The high-speed counter accumulator increments and decrements based on the quadrature relationship of the encoder's A and B inputs (I:0/0 and I:0/1). The accumulator is cleared to zero when the reset is activated or when the RES instruction is executed. All presets are entered as a relative offset to the leading edge of a manual. The presets for the hole patterns are stored in the SQO instructions. (Refer to chapter 11 for the SQO instruction.) The high-speed counter external reset input (I:0/2) and the external hold input (I:0/3) are wired in parallel to prevent the high-speed counter from counting while the reset is active.

The input filter delays for both the high-speed counter A and B inputs (I:0/0 and I:0/1) as well as the high-speed counter reset and hold inputs (I:0/2 and I:0/3) can be adjusted. Refer to page A–7 for more information on adjusting filters.

```
Rung 4:5
Interrupt occurred due to low preset reached.

|  C5:0                                          +RET---------------+-|
|----][------------------------------------------+RETURN            + |
|      IL                                        +------------------+ |
Rung 4:6
Signals the main program (file 2) to initiate a drilling sequence.  The
high-speed counter has already stopped the conveyor at the correct
position using its high preset output pattern data (clear O:0/0).  This
occurred within microseconds of the high preset being reached (just
prior to entering this high-speed counter interrupt subroutine).  The
drill sequence subroutine resets the drill sequence start bit and sets
the conveyor drive bit (O:0/0) upon completion of the drilling
sequence.

| interrupt occurred         |                   Drill Sequence Start |
| due to high preset reached |                                        |
|   C5:0                                                    B3       |
|----] [-------------------------------------------------------(L)-----|
|      IH                                                     32     |
Rung 4:7

|                                                                      |
|---------------------------------+END+-------------------------------|
|                                                                      |
```

Notes:

13 Using the Message Instruction

This chapter contains information about communications and the message (MSG) instruction. Specifically, this chapter contains information on:

- types of communication
- what the MSG instruction symbol looks like
- typical execution time for the MSG instruction
- how to use the MSG instruction
- application examples and timing diagrams

Note *Only Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers support the MSG instruction.*

Message Instruction

Instruction		Purpose	Page
Mnemonic	Name		
MSG	Message Read/Write	This instruction transfers data from one node to another via the communication port. When the instruction is enabled, the message is sent to a communication buffer. Replies are processed at the end of scan.	13–3

Types of Communication

Communication is the ability of a device to send data or status to other devices. This capability typically falls into one of two categories: initiator (master) or responder (slave). Each of these are described below:

Initiator (Master) Communication

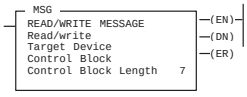
Initiator products can begin communication processes, which includes requesting information from other devices (reading) or sending information to other products (writing). In addition, initiator products are usually capable of replying to other devices when they make requests to read information. The Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers are in this class.

Initiator products can begin communication processes with other initiator products (peer-to-peer communication) or with responder (slave) products (initiator-to-responder communication).

Responder (Slave) Communication

Responder products can only reply to other products. These devices are not capable of initiating an exchange of data; they only reply to requests made from initiator products. The Series A and B MicroLogix 1000 discrete controllers are in this class.

Message Instruction (MSG)



Execution Times (µsec) when:

True	False
180①	48

The MSG is an output instruction that allows the controller to initiate an exchange of data with other devices. The relationship with the other devices can be either peer-to-peer communication or master-to-slave communication. The type of communication required by a particular application determines the programming configuration requirements of the MSG instruction.

① This only includes the amount of time needed to set up the operation requested. It does not include the time it takes to service the actual communication, as this time varies with each network configuration. As an example, 144ms is the actual communication service time for the following configuration: 3 nodes on DH-485 (2=MicroLogix 1000 programmable controllers and 1=PLC-500 A.I. Series™ programming software), running at 19.2K baud, with 2 words per transfer.

Entering Parameters

After you place the MSG instruction on a rung, specify whether the message is to be a read or write. Then specify the target device and the control block for the MSG instruction.

- **Read/Write** – read indicates that the local processor (processor in which the instruction is located) is receiving data; write indicates that the processor is sending data.
- **Target Device** – identifies the type of command used to establish communication. The target device can be a MicroLogix 1000 controller or SLC family processor using SLC commands, or a common interface file by selecting the CIF format. Valid options are:
  - SLC500/ML1000 – Allows communication between a MicroLogix 1000 controller and any other MicroLogix 1000 controller or SLC 500 family processor.
  - 485CIF – (common interface file) Allows communication between a MicroLogix 1000 controller and a non-MicroLogix 1000/SLC 500 device. The CIF data is automatically delivered to integer file 9 in SLC 500 processors or file 7 in MicroLogix 1000 controllers. The 485CIF protocol is also used for PLC-2 type messages to PLC-5 processors.

- **Control Block Address** – an integer file address that you select. It consists of 7 integer words, containing the status bits, target file address, and other data associated with the MSG instruction.
- **Control Block Length** – fixed at seven elements. This field cannot be altered.

Note

When running a MicroLogix 1000 program on an SLC 5/03 or SLC 5/04 processor, or on channel 0 of an SLC 5/05 processor, the MSG control block length increases from 7 to 14 words. If you plan to run a MicroLogix 1000 program with one of these processors, make sure that the program has at least 7 unused words following each MSG control block.

The table that follows illustrates combinations of message types and target devices and their valid file types.

Command Type	Message Type	Initiating Device	Valid File Types	Target Device①②③	Valid File Types
SLC500/ML1000	Write	MicroLogix 1000	O,I,S,B,T,C,R,N	MicroLogix 1000	O,I,S,B,T,C,R,N
SLC500/ML1000	Read	MicroLogix 1000	O,I,S,B,T,C,R,N	MicroLogix 1000	O,I,S,B,T,C,R,N
CIF	Write	MicroLogix 1000	O,I,S,B,T,C,R,N	MicroLogix 1000	N7
CIF	Read	MicroLogix 1000	O,I,S,B,T,C,R,N	MicroLogix 1000	N7
SLC500/ML1000	Write	MicroLogix 1000	O,I,S,B,T,C,R,N	SLC 500	O④,I④,S,B,T,C,R,N
SLC500/ML1000	Read	MicroLogix 1000	O,I,S,B,T,C,R,N	SLC 500	O④,I④,S,B,T,C,R,N
CIF	Write	MicroLogix 1000	O,I,S,B,T,C,R,N	SLC 500	N9
CIF	Read	MicroLogix 1000	O,I,S,B,T,C,R,N	SLC 500	N9

① The DF1 Full-Duplex protocol can be used if the target device supports it. Such devices include MicroLogix 1000 controllers (any series), SLC 5/03, SLC 5/04 and SLC 5/05 processors, and PLC-5 processors (CIF command type only).

② The DH-485 protocol can be used if the target device supports it. Such devices include MicroLogix 1000 controllers (except for Series A and B discrete controllers) and SLC 500, SLC 5/01, SLC 5/02, SLC 5/03, SLC 5/04 or SLC 5/05 processors.

③ The DF1 Half-Duplex protocol can also be used with Series D or later discrete and all analog MicroLogix 1000 controllers, but a master is required, such as an SLC 5/03, SLC 5/04 or SLC 5/05 processor.

④ SLC 500, SLC 5/01, and SLC 5/02 processors do not support O or I file access from a MSG instruction. SLC 5/03, SLC 5/04 and SLC 5/05 processors do support O and I file access, but only when unprotected.

Control Block Layout

The control block layouts shown below illustrate SLC500/ML1000 type messages.

Control Block Layout – SLC500/ML1000

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Word
EN	ST	DN	ER		EW	NR	TO	Error Code								0
								Node Number								1
Reserved for length (in elements)																2
File Number																3
File Type (O, I, S, B, T, C, R, N)																4
Element Number																5
Subelement Number																6

Control Block Layout – 485CIF

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Word
EN	ST	DN	ER		EW	NR	TO	Error Code								0
								Node Number								1
Reserved for Length (in elements)																2
Offset Bytes																3
Not used																4
Not used																5
Not used																6

Using Status Bits

```
Read/Write:                          READ          ignore if timed out:   0  TO
 Target Device:             SLC500/ML1000                to be retried:   0  NR
 Control Block:                     N7:0           awaiting execution:   0  EW
 Local Destination File Address:     ***
 Target Node:                          0                        error:   0  ER
 Target File Address:                ***                message done:   0  DN
 Message Length in elements          ***          message transmitting:   0  ST
                                                       message enabled:   0  EN

                                                   control bit address:    N7:0/8

 ERROR CODE:  0
 Error Code Desc:
```

MSG Instruction Status Bits

The right column in the display above lists the various MSG instruction status bits. These are explained below:

- **Time Out Bit TO (bit 08)** Temporarily set this bit (1) to error out (error code 37) an existing MSG instruction. This bit has no effect unless the ST bit has first been set due to receiving an ACK (an acknowlege). Your application must supply its own timer whose preset value is the MSG timeout value. This bit is reset on any false-to-true MSG rung transition.
- **Negative Response Bit NR (bit 09)** is set if the target processor is responding to your message, but can not process the message at the present time. The NR bit is reset at the next false-to-true MSG rung transition that has a transmit buffer available. It is used to determine when to send retries. The ER bit is also set at this time. Use this feedback to initiate a retry of your message at a later time. This bit is used with DH-485 protocol only.
- **Enabled and Waiting Bit EW (bit 10)** is set on any false-to-true MSG rung transition. This bit is reset when an ACK or NAK (no acknowledge) is received, or on any true-to-false MSG rung transition.

Note *The operation of the EW bit has changed since Series C.*

- **Error Bit ER (bit 12)** is set when message transmission has failed. The ER bit is reset the next time the MSG rung goes from false to true.
- **Done Bit DN (bit 13)** is set when the message is transmitted successfully. The DN bit is reset (cleared) the next time the MSG rung goes from false to true.

- **Start Bit ST (bit 14)** is set when the processor receives acknowledgement from the target device. This identifies that the target device has started to process the MSG request. The ST bit is reset when the DN, ER, or TO bit is set or on a false-to-true MSG rung transition.
- **Enable Bit EN (bit 15)** is set only if the transmit buffer is available. If the transmit buffer is not available, the EN flag remains false. When the transmit buffer becomes available, the EN flag goes true. It remains set until the next false rung execution after the MSG completes (DN bit set) or an error occurs (ER bit set).

Note *The operation of the EN bit has changed since Series C.*

The operation associated with a message read or write instruction is sent when you enable the instruction. Replies are processed at the end of the scan.

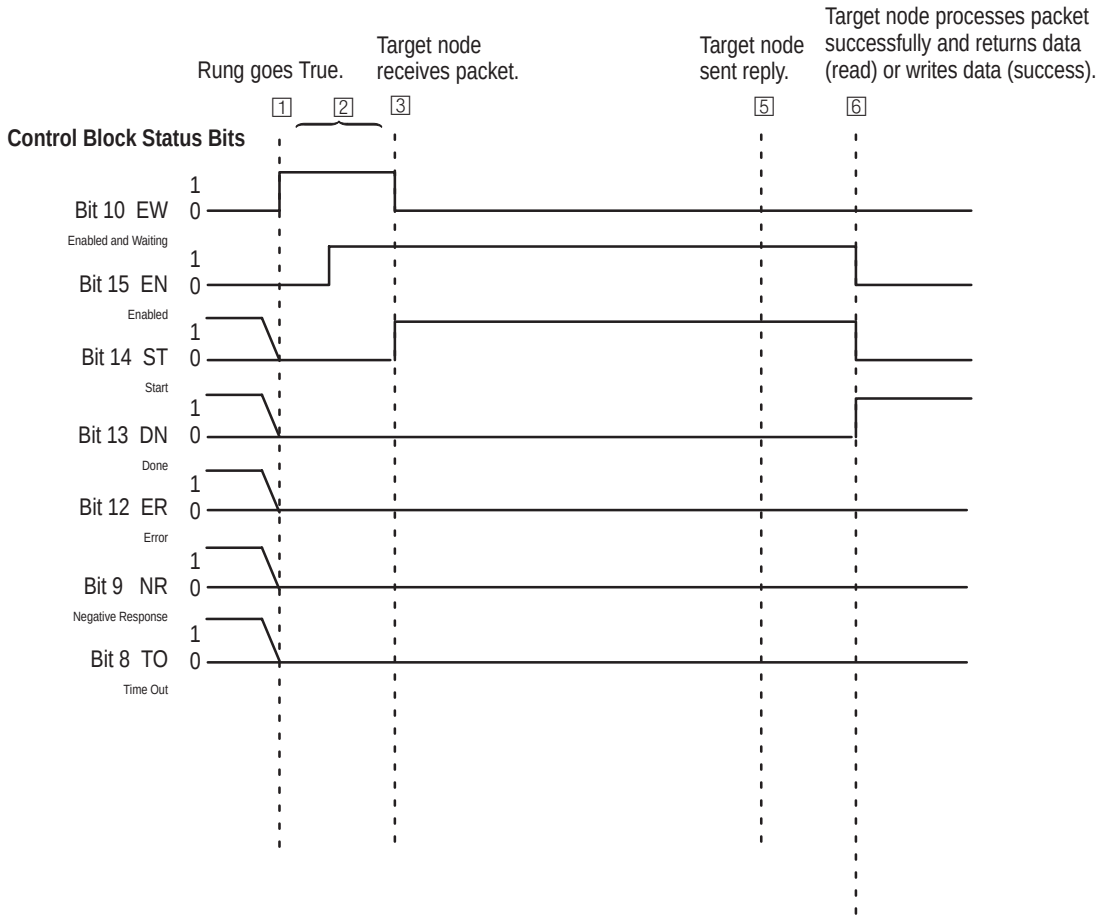
Controller Communication Status Bit

When using the MSG instruction, you should also use the following controller communication status bit:

Active Protocol Bit (S:0/11) – This is a read only bit that indicates which communication protocol is currently enabled or functioning; where 0 = DF1 (default) and 1 = DH-485. Use this bit in your program to restrict message operation to the specific protocol in use.

Timing Diagram for a Successful MSG Instruction

The following section illustrates a successful timing diagram for a Series D or later MicroLogix 1000 discrete controller, or a MicroLogix 1000 analog controller, MSG instruction.



1 The **EW** bit is set (1) and the **ST, DN, NR**, and **TO** flags are cleared. If the transmit buffer is not available, the EN flag remains false (0).

2 When rung conditions go true and the transmit buffer becomes available, the **EN** flag goes true (1). The EN bit remains set until either the **DN**, **ER**, or **TO** bit is set. The **TO** bit has no effect unless the ST bit has first been set.

3 If the Target Node successfully receives the MSG packet, it sends back an ACK (an acknowledge). The ACK causes the processor to clear bit S:2/7. (Bit S:2/7 is valid for Series C discrete only). Note that the Target Node has not yet examined the MSG packet to see if it understands your request. It is replying to the initial connection.

At the next end of scan, the **EW** bit is cleared (0) and the **ST** bit is set (1). Once the **ST** bit is set, the processor will wait indefinitely for a reply from the Target Node. The Target Node is not required to respond within any given time frame. During this time, no other MSG instruction will be serviced.

Note

*If the Target Node faults or power cycles during the time frame after the ST bit is set and before the reply is returned, you will never receive a reply. No other MSG instructions will be able to be serviced unless this MSG is terminated in error using the **TO** bit. This is why it is recommended you use a timer in conjunction with the **TO** bit to clear any pending instructions. (When the **TO** bit is set [1] it clears pending messages.) Typically message transactions are completed within a couple of seconds. It is up to the programmer to determine how long to wait before clearing the buffer and then re-transmitting.*

Step 4 is not shown in the timing diagram.

4 If you do not receive an ACK, step 3 does not occur. Instead a NAK (no acknowledge) or no response at all is received. When this happens, the **ST** bit remains clear. A NAK indicates:

- the Target Node is too busy, or
- it received a MSG packet with a bad checksum.

No response indicates:

- either the Target Node is not there, or
- it does not respond because the MSG packet was corrupted in transmission.

When a NAK occurs, the **EW** bit is cleared at the next end of scan. (Note that the NR bit will only be set for DH-485 and NAK conditions. An error code 02H, Target Node is busy, is received which causes the **NR** bit to be set.) The **ER** bit is also set which indicates that the MSG instruction failed.

Monitor the **NR** bit. If it is set, indicating that the Target Node is busy, you may want to initiate some other process (e.g., an alarm or a retry later). The **NR** bit is cleared when the rung logic preceding the MSG changes from false to true.

5 When an ACK occurs, the Target Node sends one of three responses shown in Step 6.

6. Following the successful receipt of the packet, the Target Node sends a reply packet. The reply packet will contain one of the following responses:

   - I have successfully performed your write request.
   - I have successfully performed your read request, and here is your data.
   - I have not performed your request because of an error.

   At the next end of scan, following the Target Node's reply, the MicroLogix 1000 controller examines the MSG packet from the target device. If the reply contains "I have successfully performed your write request," the **DN** bit is set and the **ST** bit is cleared. The MSG instruction is complete. If the MSG rung is false, the **EN** bit is cleared the next time the MSG instruction is scanned.

   If the reply contains "I have successfully performed your read request, and here is your data," the data is written to the appropriate data table, the **DN** bit is set, and the **ST** bit is cleared. The MSG instruction function is complete. If the MSG rung is false, the **EN** bit is cleared the next time the MSG instruction is scanned.

   If the reply contains "I have not performed your request, because of an error," the **ER** bit is set and the **ST** bit is cleared. The MSG instruction function is complete. If the MSG rung is false, the **EN** bit is cleared the next time the MSG instruction is scanned.

MSG Instruction Error Codes

Note *Any MSG instruction that is in progress during a network protocol switch will not be processed and will be discarded. For more information on network protocol switching, see page 3–17.*

When an error condition occurs, the error code is stored in the *lower byte* of the first control word assigned to the MSG instruction.

Error Code	Description of Error Condition
02H	Target node is busy.
03H	Target node cannot respond because message is too large.
04H	Target node cannot respond because it does not understand the command parameters OR the control block may have been inadvertently modified.
05H	Local processor is offline (possible duplicate node situation).
06H	Target node cannot respond because requested function is not available.
07H	Target node does not respond.
08H	Target node cannot respond.
09H	Local modem connection has been lost.
0AH	Buffer unavailable to receive SRD reply.
0BH	Target node does not accept this type of MSG instruction.
0CH	Received a master link reset.
10H	Target node cannot respond because of incorrect command parameters or unsupported command.
15H	Local channel configuration parameter error exists.
18H	Broadcast (Node Address 255) is not supported.
1AH①	Target node cannot respond because another node is file owner (has sole file access).
1BH①	Target node cannot respond because another node is program owner (has sole access to all files).
37H	Message timed out in local processor.
39H	Message was discarded due to a communication protocol switch.
3AH	Reply from target is invalid.
50H	Target node is out of memory.
60H	Target node cannot respond because file is protected.
E7H	Target node cannot respond because length requested is too large.
EBH	Target node cannot respond because target node denies access.
ECH	Target node cannot respond because requested function is currently unavailable.
FAH	Target node cannot respond because another node is file owner (has sole file access).
FBH	Target node cannot respond because another node is program owner (has sole access to all files).

① Error codes 1A and 1B valid for Series C discrete only.

Note

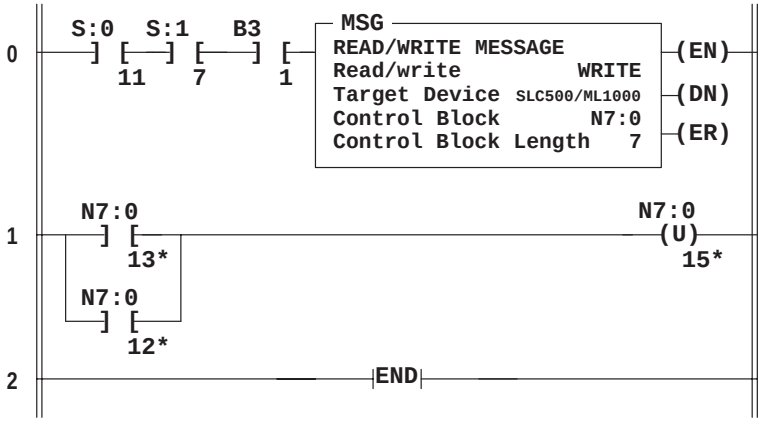
For 1770–6.5.16 DF1 Protocol and Command Set users:

The MSG error code reflects the STS field of the reply to your MSG instruction. Codes E0 – EF represent EXT STS codes 0 – F. Codes F0 – FC represent EXT STS codes 10 – 1C.

Application Examples that Use the MSG Instruction

Example 1

Application example 1 shows how you can implement continuous operation of a message instruction.



* MSG instruction
status bits:
12 = ER
13 = DN
15 = EN

Operation Notes

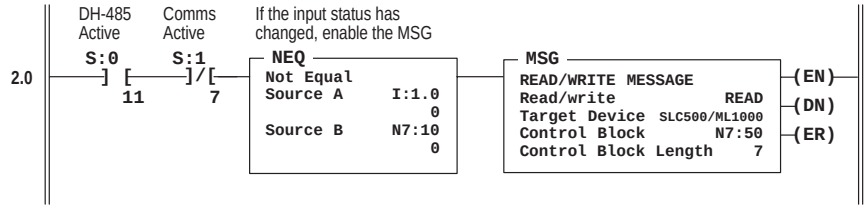
Bit S:0/11 ensures that the MSG instruction will only be processed when the active protocol is DH-485. Bit S:1/7 ensures that DH-485 is communicating before sending the MSG. Bit B3/1 enables the MSG instruction. When the MSG instruction done bit (N7:0/13) is set, it unlatches the MSG enable bit (N7:0/15) so that the MSG instruction will be re-enabled in the next scan. This provides continuous operation.

The MSG error bit will also unlatch the enable bit. This provides continuous operation even if an error occurs.

Example 2

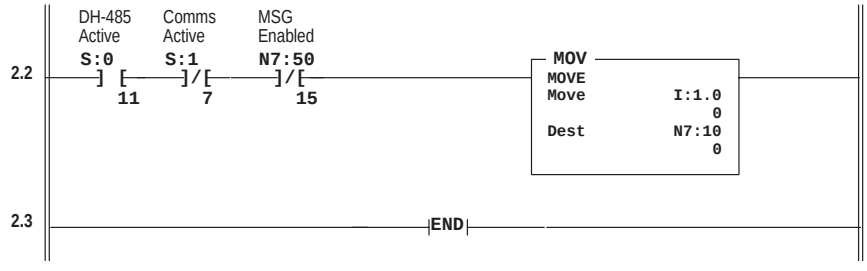
Application example 2 involves a MicroLogix 1000 controller transmitting its first input word to another MicroLogix 1000 controller. This is commonly referred to as "change of state" or "report on exception" messaging. Using this type of logic significantly reduces network traffic, which in turn significantly improves network throughput.

This is the message control rung. The logic preceding the MSG instruction on this rung dictates when the MSG instruction is processed. In this example, the MSG instruction will only be processed when the active protocol is DH-485 and when there is no other communication. Once the MSG instruction is enabled, it locks itself into operation regardless of the preceding logic on the rung.



This rung controls when the MSG instruction is unlatched or reset. The MSG instruction must be reset before it can re-transmit new information. Either of the following two conditions will reset the MSG instruction: 1) when communication to the target device have been completed successfully, or 2) when an error is detected in the communication sequence (occurs after all retries have been exhausted). Using the error bit to reset the MSG is primarily used to stop the MSG instruction from being totally locked out.

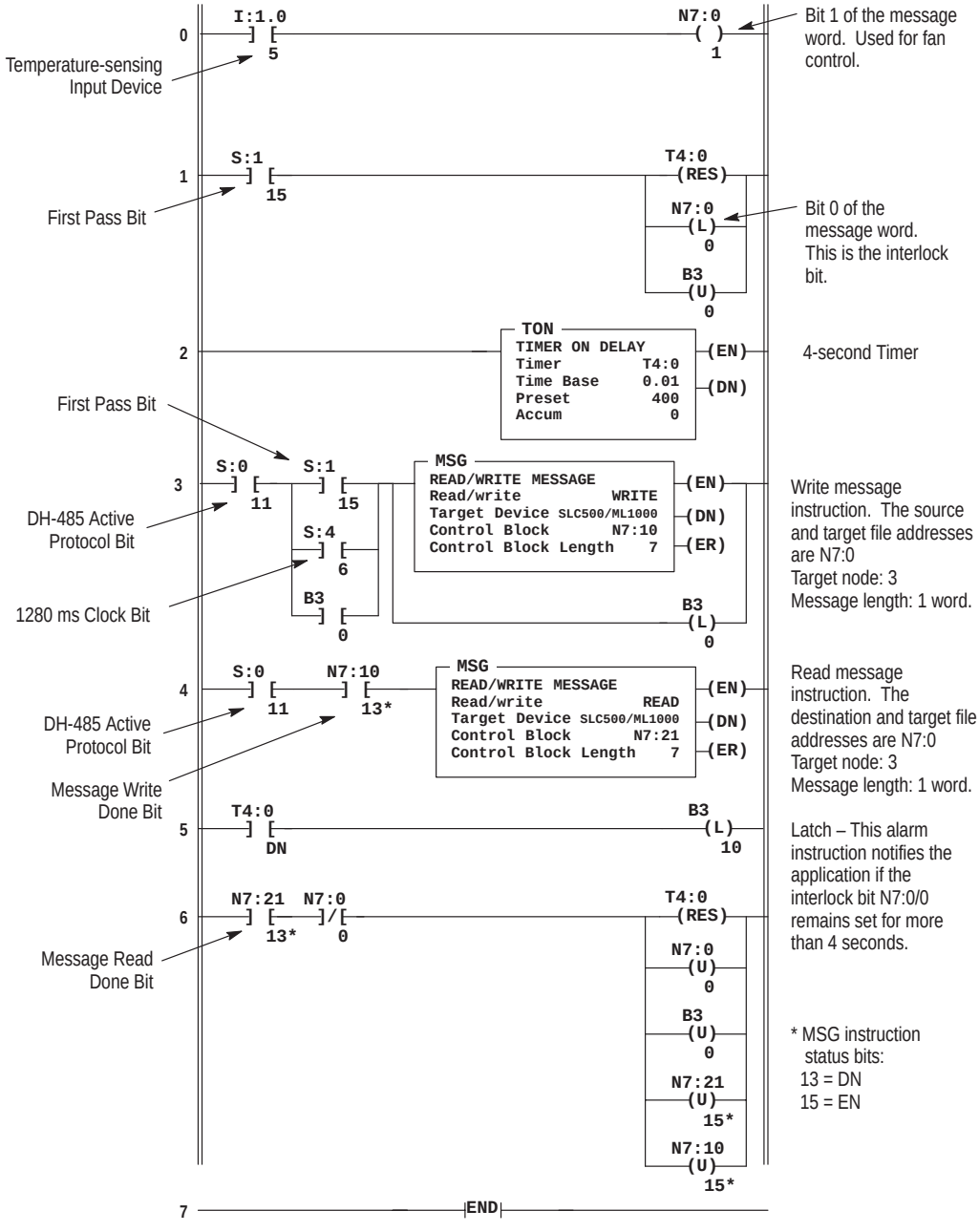
This rung is used to setup the "report by exception" operation. This move command updates N7:10, by making it identical to I:1.0. When the processor starts a new scan sequence (when rung 2.0 is scanned,) it updates (reads) the input image. If an input has changed from the previous scan, the NEQ instruction will be true and MSG will be processed. The MSG Enabled bit ensures that the MOV will not be processed until after the MSG is successfully completed. This minimizes the chances that input changes are missed during MSG operation.



Example 3

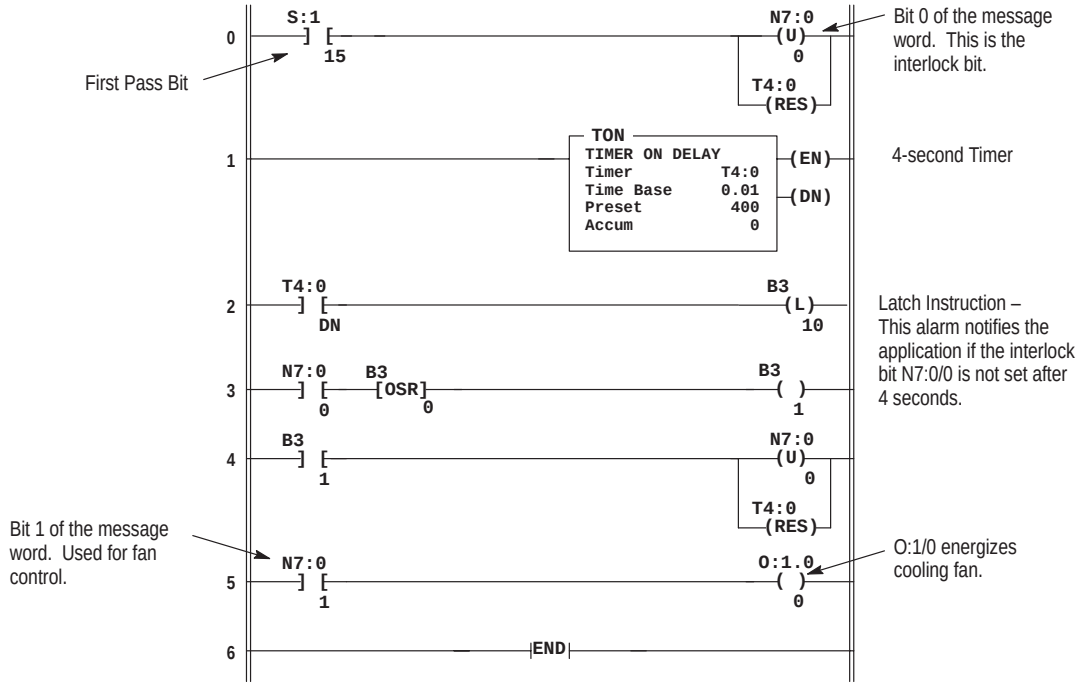
Application example 3 involves a MicroLogix 1000 controller and an SLC 5/01 processor communicating on a DH-485 network. Interlocking is provided to verify data transfer and to shut down both processors if communication fails.

A temperature-sensing device, connected as an input to the MicroLogix 1000 controller, controls the on-off operation of a cooling fan, connected as an output to the SLC 5/01 processor. The MicroLogix 1000 and SLC 5/01 ladder programs are explained on the following pages.



Operation notes appear on the following page.

Program File 2 of SLC 5/01 Processor at Node 3



Operation Notes, MicroLogix 1000 and SLC 5/01 programs

Message instruction parameters: N7:0 is the message word. It is the target file address (SLC 5/01 processor) and the local source and destination addresses (MicroLogix 1000 controller) in the message instructions.

N7:0/0 of the message word is the interlock bit; it is written to the 5/01 processor as a 1 (set) and read from the SLC 5/01 processor as a 0 (reset).

N7:0/1 of the message word controls cooling fan operation; it is written to the SLC 5/01 processor as a 1 (set) if cooling is required or as a 0 (reset) if cooling is not required. It is read from the SLC 5/01 processor as either 1 or 0.

Word N7:0 should have a value of 1 or 3 during the message write execution. N7:0 should have a value of 0 or 2 during the message read execution.

Program initialization: The first pass bit S:1/15 initializes the ladder programs on run mode entry.

MicroLogix 1000 controller: N7:0/0 is latched; timer T4:0 is reset; B3/0 is unlatched (rung 1), then latched (rung 3).
SLC 5/01 processor: N7:0/0 is unlatched; timer T4:0 is reset.

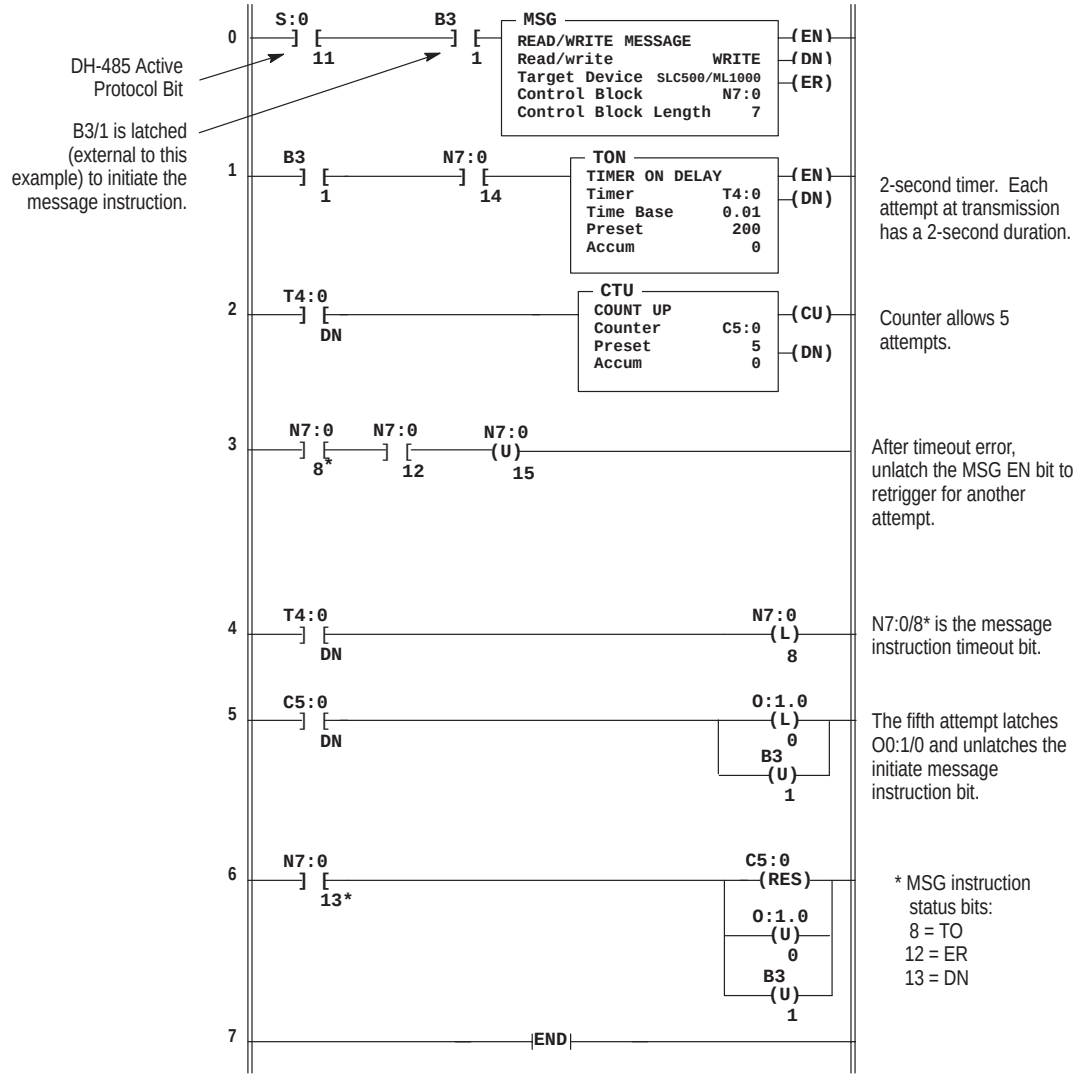
Message instruction operation: The message write instruction in the MicroLogix 1000 controller is initiated every 1280 ms by clock bit S:4/6. The done bit of the message write instruction initiates the message read instruction.

B3/0 latches the message write instruction. B3/0 is unlatched when the message read instruction done bit is set, provided that the interlock bit N7:0/0 is reset.

Communication failure: In the MicroLogix 1000 controller, bit B3/10 becomes set if interlock bit N7:0/0 remains set (1) for more than 4 seconds. In the SLC 5/01 processor, bit B3/10 becomes set if interlock bit N7:0/0 remains set (1) for more than 4 seconds. Your application can detect this event, take appropriate action, then unlatch bit B3/10.

Example 4

Application example 4 shows you how to use the timeout bit to disable an active message instruction. In this example, an output is energized after five unsuccessful attempts (two seconds duration) to transmit a message.



Operation Notes

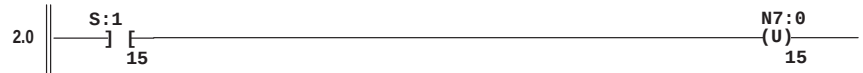
The timeout bit is latched (rung 4) after a period of 2 seconds. This clears the message instruction from processor control on the next scan. The message instruction is then re-enabled for a second attempt at transmission. After 5 attempts, O:1/0 is latched and B3/1 is unlatched.

A successful attempt at transmission resets the counter, unlatches O:1/0, and unlatches B3/1.

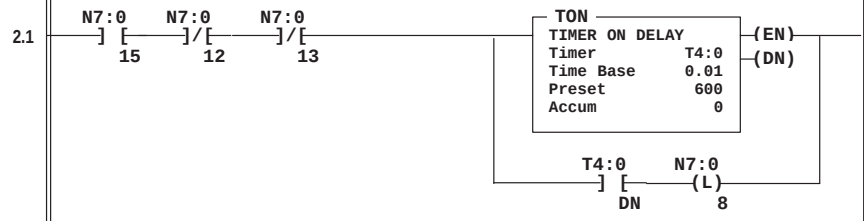
Example 5

Application example 5 shows you how to link message instructions together to transmit serially, one after another. In this example a MSG Write is followed by a MSG Read which causes the serial transmission.

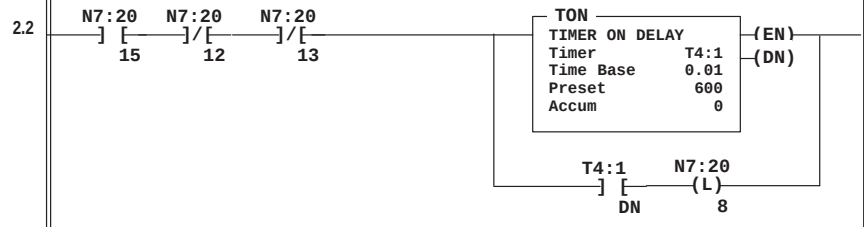
This rung starts messaging each REM Run or RUN mode entry by clearing the EN bit of the first MSG instruction.



This rung sets the timeout value. (When using a SLC 5/03 or SLC 5/04 processor, this rung and rung 2:2 are not required because you can enter the value 6 into the Timeout value field in the MSG instruction block.)

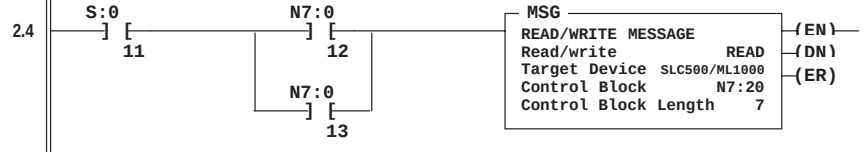


Same as above rung.

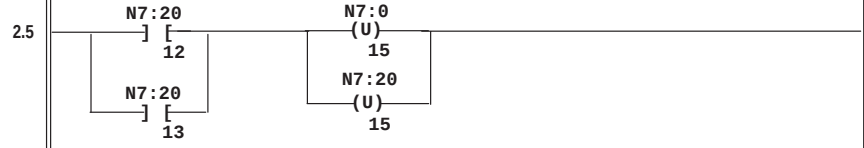


The MSG instruction energizes upon entry into the REM Run or RUN mode. No input conditions are required.

The MSG instruction is energized when the previous MSG instruction completes.



This rung resets all MSG instructions when the last MSG instruction has completed.



Notes:

14 Troubleshooting Your System

This chapter describes how to troubleshoot your controller. Topics include:

- understanding the controller LED status
- controller error recovery model
- identifying controller faults
- calling Allen-Bradley for assistance

Understanding the Controller LED Status

Between the time you apply power to the controller and the time it has to establish communication with a connected programming device, the only form of communication between you and the controller is through the LEDs.

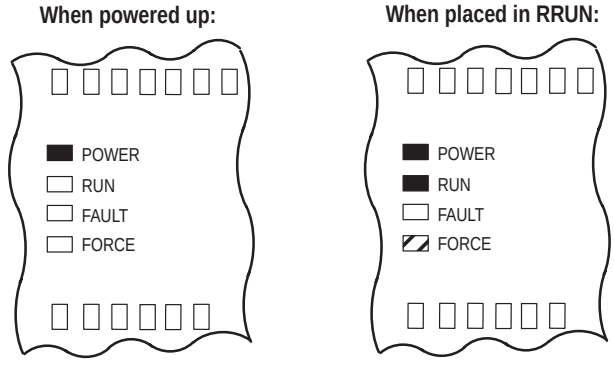
When Operating Normally

When power is applied, only the power LED turns on and remains on. This is part of the normal powerup sequence.

When the controller is placed in REM Run mode, the run LED also turns on and remains on, as shown on the right in the figure below. If a force exists, the force LED is on as well.

Refer to the following key to determine the status of the LED indicators:

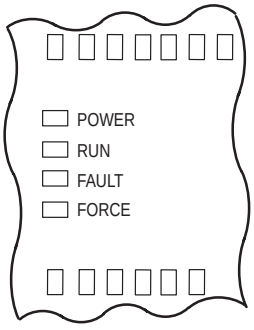
- Indicates the LED is OFF.
- Indicates the LED is ON.
- Indicates the LED is FLASHING.
- Status of LED does not matter.



When an Error Exists

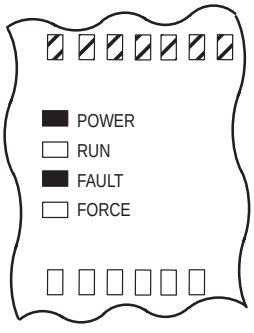
If an error exists within the controller, the controller LEDs operate as described in the following tables.

If the LEDs indicate:



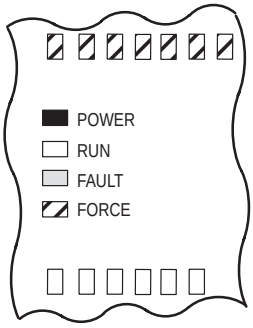
The Following Error Exists	Probable Cause	Recommended Action
No input power or power supply error	No Line Power	Verify proper line voltage and connections to the controller.
	Power Supply Overloaded	This problem can occur intermittently if power supply is overloaded when output loading and temperature varies.

If the LEDs indicate:



The Following Error Exists	Probable Cause	Recommended Action
Hardware faulted	Processor Memory Error	Cycle power. Contact your local Allen-Bradley representative if the error persists.
	Loose Wiring	Verify connections to the controller.

If the LEDs indicate:



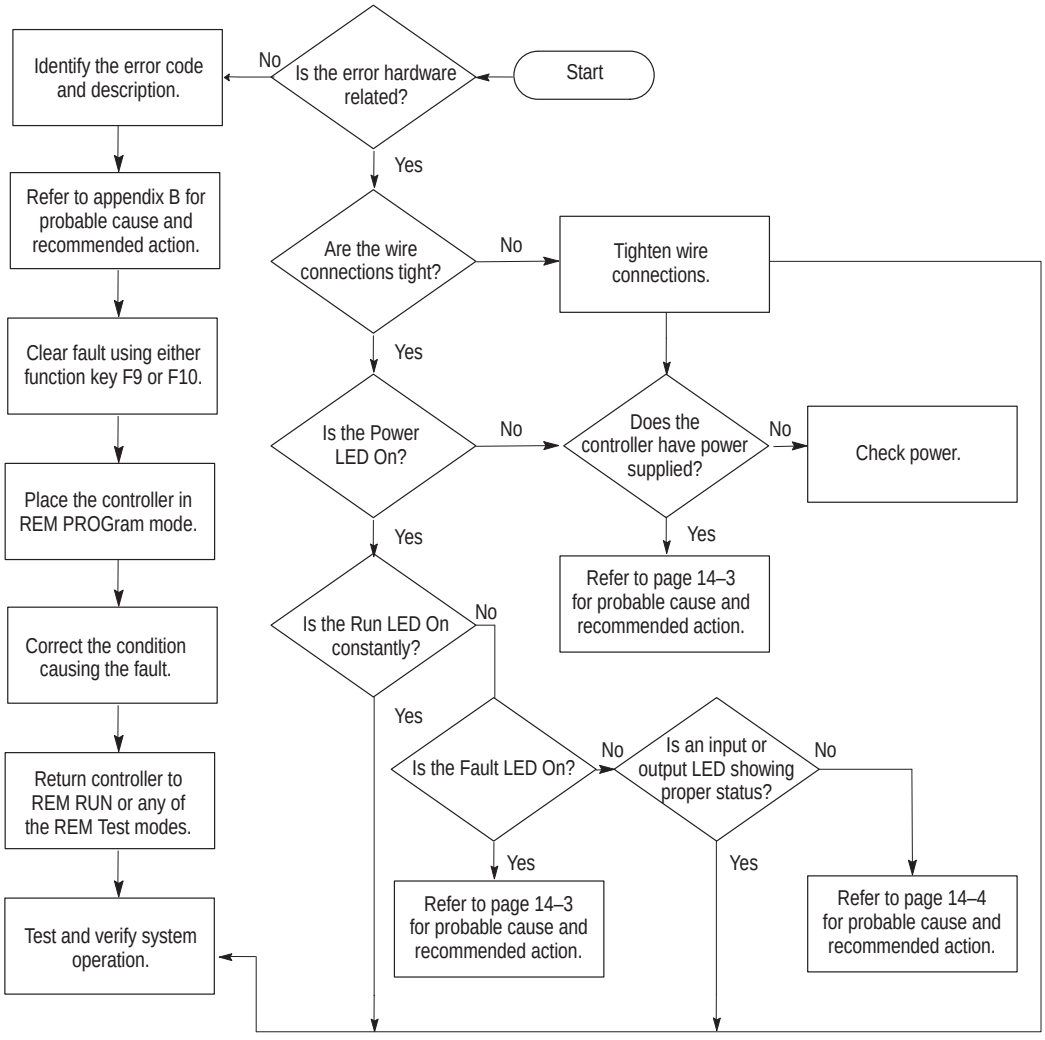
The Following Error Exists	Probable Cause	Recommended Action
Application fault	Hardware/Software Major Fault Detected	**1.** Monitor Status File Word S:6 for major error code. **2.** Remove hardware/software condition causing fault. **3.** Press F10 to clear the fault. **4.** Attempt a controller REM Run mode entry. If unsuccessful, repeat recommended action steps above or contact your local Allen-Bradley distributor.

Refer to the following key to determine the status of the LED indicators:

- Indicates the LED is OFF.
- Indicates the LED is ON.
- Indicates the LED is FLASHING.
- Status of LED does not matter.

Controller Error Recovery Model

Use the following error recovery model to help you diagnose software and hardware problems in the micro controller. The model provides common questions you might ask to help troubleshoot your system. Refer to the recommended pages within the model and to S:6 of the status file on page B–14 for further help.


Start
Is the error hardware related?
No
Identify the error code and description.
Refer to appendix B for probable cause and recommended action.
Clear fault using either function key F9 or F10.
Place the controller in REM PROGram mode.
Correct the condition causing the fault.
Return controller to REM RUN or any of the REM Test modes.
Test and verify system operation.
Yes
Are the wire connections tight?
No
Tighten wire connections.
Yes
Is the Power LED On?
No
Does the controller have power supplied?
No
Check power.
Yes
Refer to page 14–3 for probable cause and recommended action.
Yes
Is the Run LED On constantly?
No
Yes
Is the Fault LED On?
No
Is an input or output LED showing proper status?
No
Yes
Refer to page 14–3 for probable cause and recommended action.
Yes
Refer to page 14–4 for probable cause and recommended action.

Identifying Controller Faults

While a program is executing, a fault may occur within the operating system or your program. When a fault occurs, you have various options to determine what the fault is and how to correct it. This section describes how to clear faults and provides a list of possible advisory messages with recommended corrective actions.

Automatically Clearing Faults

You can automatically clear a fault when cycling power to the controller by setting either one or both of the following status bits in the status file:

- Fault Override at Powerup bit (S:1/8)
- Run Always bit (S:1/12)

Clearing a fault using the Run Always bit (S:1/12) causes the controller to immediately enter the REM Run mode. Make sure you fully understand the use of this bit before incorporating it into your program. Refer to page B–6 for more information.

Refer to appendix B for more information on status bits.

Note *You can declare your own application-specific major fault by writing your own unique value to S:6 and then setting bit S:1/13 to prevent reusing system defined codes. The recommended values for user defined faults is FF00 to FF0F.*

Manually Clearing Faults Using the Fault Routine

The occurrence of recoverable or non-recoverable user faults causes file 3 to be executed. If the fault is recoverable, the subroutine can be used to correct the problem and clear the fault bit S:1/13. The controller then continues in the REM Run mode.

The subroutine does not execute for non-user faults. The user-fault routine is discussed in chapter 4.

Fault Messages

This section contains fault messages that can occur during operation of the MicroLogix 1000 programmable controllers. Each table lists the error code description, the probable cause, and the recommended corrective action.

Error Code (Hex)	Advisory Message	Description	Recommended Action
0001	DEFAULT PROGRAM LOADED	The default program is loaded to the controller memory. This occurs: • on power up if the power down occurred in the middle of a download • if the user program is corrupt at power up, the default program is loaded.	• Re-download the program and enter the REM Run mode. • Contact your local Allen-Bradley representative if the error persists.
0002	UNEXPECTED RESET	The controller was unexpectedly reset due to a noisy environment or internal hardware failure. If the user program downloaded to the controller is valid, the initial data downloaded with the program is used. The Retentive Data Lost Bit (S:5/8) is set. If the user program is invalid, the default program is loaded.	• Refer to proper grounding guidelines in chapter 2. • Contact your local Allen-Bradley representative if the error persists.
0003	EEPROM MEMORY IS CORRUPT	While power cycling to your controller, a noise problem may have occurred. Your program may be valid, but retentive data will be lost.	• Try cycling power again. • Contact your local Allen-Bradley representative if the error persists.
0004	RUNTIME MEMORY INTEGRITY ERROR	While the controller was in the RUN mode or any test mode, the ROM or RAM became corrupt. If the user program is valid, the program and initial data downloaded to the controller is used and the Retentive Data Lost Bit (S:5/8) is set. If the user program is invalid, error 0003 occurs.	• Cycle power on your unit. • Download your program and re-initialize any necessary data. • Start up your system. • Contact your local Allen-Bradley representative if the error persists.
0005	RETENTIVE DATA HAS BEEN LOST	The data files (input, output, timer, counter, integer, binary, control, and status) are corrupt.	• Cycle power on your unit. • Download your program and re-initialize any necessary data. • Start up your system. • Contact your local Allen-Bradley representative if the error persists.
0008	FATAL INTERNAL SOFTWARE ERROR	The controller software has detected an invalid condition within the hardware or software after completing power-up processing (after the first 2 seconds of operation).	• Cycle power on your unit. • Download your program and re-initialize any necessary data. • Start up your system. • Contact your local Allen-Bradley representative if the error persists.

Error Code (Hex)	Advisory Message	Description	Recommended Action
0009	FATAL INTERNAL HARDWARE ERROR	The controller software has detected an invalid condition within the hardware during power-up processing (within the first 2 seconds of operation).	• Cycle power on your unit. • Download your program and re-initialize any necessary data. • Start up your system. • Contact your local Allen-Bradley representative if the error persists.
0010	INCOMPATIBLE PROCESSOR	The downloaded program is not configured for a micro controller.	If you want to use a micro controller with the program, reconfigure your controller with your programming software (choose Bul. 1761).
0016	STARTUP PROTECTION AFTER POWERLOSS; S:1/9 IS SET	The system has powered up in the REM Run mode. Bit S:1/13 is set and the user-fault routine is run before beginning the first scan of the program.	• Either reset bit S:1/9 if this is consistent with your application requirements, and change the mode back to REM Run, or • clear S:1/13, the major fault bit.
0018	USER PROGRAM IS INCOMPATIBLE WITH OPERATING SYSTEM	An incompatible program was downloaded. Either the program does not have the correct number of files or it does not have the correct size data files. The default program is loaded.	• Check the configuration and make sure the correct processor is selected. • If you want to use a micro controller with the program, reconfigure your controller with your programming software (choose Bul. 1761).
0020	MINOR ERROR AT END OF SCAN, SEE S:5	A minor fault bit (bits 0-7) in S:5 was set at the end of scan.	• Enter the status file display and clear the fault. • Return to the REM Run mode.
0022	WATCHDOG TIMER EXPIRED, SEE S:3	The program scan time exceeded the watchdog timeout value (S:3H).	• Verify if the program is caught in a loop and correct the problem. • Increase the watchdog timeout value in the status file.
0024	INVALID STI INTERRUPT SETPOINT, SEE S:30	An invalid STI interval exists (not between 0 and 255).	Set the STI interval between the values of 0 and 255.
0025	TOO MANY JSRs IN STI SUBROUTINE	There are more than 3 subroutines nested in the STI subroutine (file 5).	• Correct the user program to meet the requirements and restrictions for the JSR instruction. • Reload the program and enter the REM Run mode.
0027	TOO MANY JSRs IN FAULT SUBROUTINE	There are more than 3 subroutines nested in the fault routine (file 3).	• Correct the user program to meet the requirements and restrictions for the JSR instruction. • Reload the program and enter the REM Run mode.
002A	INDEXED ADDRESS TOO LARGE FOR FILE	The program is referencing through indexed addressing an element beyond a file boundary.	Correct the user program to not go beyond file boundaries.

Error Code (Hex)	Advisory Message	Description	Recommended Action
002B	TOO MANY JSRs IN HSC	There are more than 3 subroutines nested in the high-speed counter routine (file 4).	• Correct the user program to meet the requirements and restrictions for the JSR instruction. • Reload the program and enter the REM Run mode.
0030	SUBROUTINE NESTING EXCEEDS LIMIT OF 8	There are more than 8 subroutines nested in the main program file (file 2).	• Correct the user program to meet the requirements and restrictions for the main program file. • Reload the program and enter the REM Run mode.
0031	UNSUPPORTED INSTRUCTION DETECTED	The program contains an instruction(s) that is not supported by the micro controller. For example SVC or PID.	• Modify the program so that all instructions are supported by the controller. • Reload the program and enter the REM Run mode.
0032	SQO/SQC CROSSED DATA FILE BOUNDARIES	A sequencer instruction length/position parameter points past the end of a data file.	• Correct the program to ensure that the length and position parameters do not point past the data file. • Reload the program and enter the REM Run mode.
0033	BSL/BSR/FFL/FFU/LFL/LFU CROSSED DATA FILE BOUNDARIES	The length parameter of a BSL, BSR, FFL, FFU, LFL, or LFU instruction points past the end of a data file.	• Correct the program to ensure that the length parameter does not point past the data file. • Reload the program and enter the REM Run mode.
0034	NEGATIVE VALUE IN TIMER PRESET OR ACCUMULATOR	A negative value was loaded to a timer preset or accumulator.	• If the program is moving values to the accumulated or preset word of a timer, make certain these values are not negative. • Reload the program and enter the REM Run mode.
0035	ILLEGAL INSTRUCTION (TND) IN INTERRUPT FILE	The program contains a Temporary End (TND) instruction in file 3, 4, or 5 when it is being used as an interrupt subroutine.	• Correct the program. • Reload the program and enter the REM Run mode.
0037	INVALID PRESETS LOADED TO HIGH-SPEED COUNTER	Either a zero (0) or a negative high preset was loaded to counter (C5:0) when the HSC was an Up counter or the high preset was lower than or equal to the low preset when the HSC was a Bidirectional counter.	• Check to make sure the presets are valid. • Correct the program, reload, and enter the REM Run mode.
0038	SUBROUTINE RETURN INSTRUCTION (RET) IN PROGRAM FILE 2	A RET instruction is in the main program file (file 2).	• Remove the RET instruction. • Reload the program and enter the REM Run mode.

Error Code (Hex)	Advisory Message	Description	Recommended Action
0040	OUTPUT VERIFY WRITE FAILURE	When outputs were written and read back by the controller, the read failed. This may have been caused by noise.	• Refer to proper grounding guidelines in chapter 2. • Start up your system. • Contact your local Allen-Bradley representative if the error persists.
0041[1]	EXTRA OUTPUT BIT(S) TURNED ON	An extra output bit was set when the Extra Output Select (S:0/8) bit in the status file was reset. For 16-point controllers this includes bits 6–15. For 32-point controllers this includes bits 12–15.	• Set S:0/8 or change your application to *prevent* these bits from being turned on. • Correct the program, reload, and enter the REM Run mode.

[1] Valid for Series A – C discrete only.

Calling Allen-Bradley for Assistance

If you need to contact Allen-Bradley or local distributor for assistance, it is helpful to obtain the following (prior to calling):

- controller type, series letter, firmware (FRN) number (on controller's side label)
- controller LED status
- controller error codes (found in S:6 of status file)

A Hardware Reference

This appendix lists the controller:

- specifications
- dimensions
- replacement parts

For AIC+ specifications, see the *Advanced Interface Converter (AIC+) and DeviceNet Interface (DNI) Installation Instructions,* Publication 1761-5.11.

Controller Specifications

Controller Types

Catalog Number	Description
1761-L16AWA	10 pt. ac input, 6 pt. relay output, ac power supply controller
1761-L32AWA	20 pt. ac input, 12 pt. relay output, ac power supply controller
1761-L20AWA-5A	12 pt. ac input, 4 pt. analog input, 8 pt. relay output, 1 pt. analog output, ac power supply controller
1761-L10BWA	6 pt. dc input, 4 pt. relay output, ac power supply controller
1761-L16BWA	10 pt. dc input, 6 pt. relay output, ac power supply controller
1761-L20BWA-5A	12 pt. dc input, 4 pt. analog input, 8 pt. relay output, 1 pt. analog output, ac power supply controller
1761-L32BWA	20 pt. dc input, 12 pt. relay output, ac power supply controller
1761-L10BWB	6 pt. dc input, 4 pt. relay output, dc power supply controller
1761-L16BWB	10 pt. dc input, 6 pt. relay output, dc power supply controller
1761-L20BWB-5A	12 pt. dc input, 4 pt. analog input, 8 pt. relay output, 1 pt. analog output, dc power supply controller
1761-L32BWB	20 pt. dc input, 12 pt. relay output, dc power supply controller
1761-L16BBB	10 pt. dc input, 4 pt. FET and 2 pt. relay outputs, dc power supply controller
1761-L32BBB	20 pt. dc input, 10 pt. FET and 2 pt. relay outputs, dc power supply controller
1761-L32AAA	20 pt. ac input, 10 pt. triac and 2 pt. relay outputs, ac power supply controller

General Specifications

Description:		Specification: 1761-L													
		16AWA	20AWA-5A	32AWA	10BWA	16BWA	20BWA-5A	32BWA	32AAA	16BBB	10BWB	16BWB	20BWB-5A	32BWB	32BBB
Memory Size/Type		1 K EEPROM (approximately 737 instruction words: 437 data words)													
Power Supply Voltage		85–264V ac, 47-63 Hz								20.4–26.4V dc					
Power Supply Usage	120V ac	15 VA	20 VA	19 VA	24 VA	26 VA	30 VA	29 VA	16 VA	Not Applicable					
	240V ac	21 VA	27 VA	25 VA	32 VA	33 VA	38 VA	36 VA	22 VA						
	24V dc	Not Applicable								5W			10W	7W	
Power Supply Max. Inrush Current①		30A for 8 ms								30A for 4 ms			50A for 4 ms	30A for 4 ms	
24V dc Sensor Power (V dc at mA)		Not Applicable			200 mA				Not Applicable						
Max Capacitive Load (User 24V dc)					200 µF										
Power Cycles		50,000 minimum													
Operating Temp.		Horizontal mounting: 0°C to +55°C (+32°F to +131°F) for horizontal mounting Vertical mounting②: 0°C to +45°C (+32°F to +113°F) for discrete; 0°C to +40°C (+32°F to +113°F) for analog													
Storage Temp.		–40°C to +85°C (–40°F to +185°F)													
Operating Humidity		5 to 95% noncondensing													
Vibration		Operating: 5 Hz to 2k Hz, 0.381 mm (0.015 in.) peak to peak/2.5g panel mounted,③ 1hr per axis Non-operating: 5 Hz to 2k Hz, 0.762 mm (0.030 in.) peak to peak/5g, 1hr per axis													
Shock④		Operating: 10g peak acceleration (7.5g DIN rail mounted)⑤ (11±1 ms duration) 3 times each direction, each axis Non-operating: 20g peak acceleration (11±1 ms duration), 3 times each direction, each axis													
Agency Certification (when product or packaging is marked)		• C-UL Class I, Division 2 Groups A, B, C, D certified • UL listed (Class I, Division 2 Groups A, B, C, D certified) • CE marked for all applicable directives													
Terminal Screw Torque		0.9 N-m maximum (8.0 in.-lbs)													
Electrostatic Discharge		IEC801-2 @ 8K V Discrete I/O 4K V Contact, 8K V Air for Analog I/O													
Radiated Susceptibility		IEC801-3 @ 10 V/m, 27 MHz – 1000 MHz except for 3V/m, 87 MHz – 108 MHz, 174 MHz – 230 MHz, and 470 MHz – 790 MHz													
Fast Transient		IEC801-4 @ 2K V Power Supply, I/O; 1K V Comms													
Isolation		1500V ac													

① Refer to page 1–13 for additional information on power supply inrush.

② DC input voltage derated linearly from 30°C (30V to 26.4V).

③ DIN rail mounted controller is 1g.

④ Refer to page 1–18 for vertical mounting specifications.

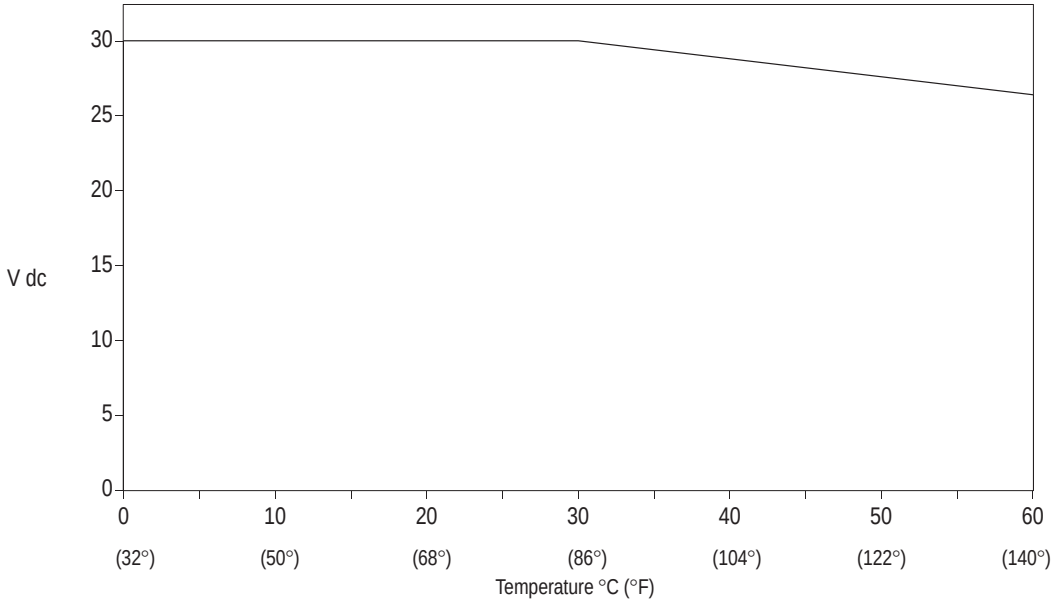
⑤ Relays are derated an additional 2.5g on 32 pt. controllers.

Input Specifications

Description	Specification	
	100-120V ac Controllers	24V dc Controllers
Voltage Range	79 to132V ac 47 to 63 Hz	14 to 30V dc
On Voltage	79V ac min. 132V ac max.	14V dc min. 24V dc nominal 26.4V dc max. @ 55°C (131°F) 30.0V dc max. @ 30°C (86°F)
Off Voltage	20V ac	5V dc
On Current	5.0 mA min. @ 79V ac 47 Hz 12.0 mA nominal @ 120V ac 60 Hz 16.0 mA max. @ 132V ac 63 Hz	2.5 mA min. @ 14V dc 8.0 mA nominal @ 24V dc 12.0 mA max. @ 30V dc
Off Current	2.5 mA max.	1.5 mA max.
Nominal Impedance	12K ohms @ 50 Hz 10K ohms @ 60 Hz	3K ohms
Inrush Maximum	250 mA max.①	Not Applicable

① To reduce the inrush maximum to 35 mA, apply a 6.8K ohm, 5w resistor in series with the input. The on-state voltage increases to 92V ac as a result.

dc Input Derating Graph



General Output Specifications

Type	Relay	MOSFET	Triac
Voltage	See Wiring Diagrams, p. 2–7.		
Maximum Load Current	Refer to the Relay Contact Rating Table.	1.0A per point @ 55° C (131° F) 1.5A per point @ 30° C (86° F)	0.5A per point @ 55° C (131° F) 1.0A per point @ 30° C (86° F)
Minimum Load Current	10.0 mA	1 mA	10.0 mA
Current per Controller	1440 VA	3A for L16BBB 6A for L32BBB	1440 VA
Current per Common	8.0A	3A for L16BBB 6A for L32BBB	Not Applicable
Maximum Off State Leakage Current	0 mA	1 mA	2 mA @ 132V ac 4.5 mA @ 264V ac
Off to On Response	10 ms max.	0.1 ms	8.8 ms @ 60 Hz 10.6 ms @ 50 Hz
On to Off Response	10 ms max.	1 ms	11.0 ms
Surge Current per Point	Not Applicable	4A for 10 ms①	10A for 25 ms①

① Repeatability is once every 2 seconds at 55° C (131° F).

Relay Contact Rating Table (applies to all Bulletin 1761 controllers)

Maximum Volts	Amperes		Amperes Continuous per Point	Voltamperes	
	Make	Break		Make	Break
240V ac	7.5A	0.75A	2.5A	1800 VA	180 VA
120V ac	15A	1.5A			
125V dc	0.22A①		1.0A	28 VA	
24V dc	1.2A①		2.0A	28 VA	

① For dc voltage applications, the make/break ampere rating for relay contacts can be determined by dividing 28 VA by the applied dc voltage. For example, 28 VA ÷ 48V dc = 0.58A. For dc voltage applications less than 48V, the make/break ratings for relay contacts cannot exceed 2A. For dc voltage applications greater than 48V, the make/break ratings for relay contacts cannot exceed 1A.

Analog Input Specifications

Description	Specification
Voltage Input Range	–10.5 to +10.5V dc – 1LSB
Current Input Range	–21 to +21 mA – 1LSB
Type of Data	16-bit signed integer
Input Coding –21 to +21 mA – 1LSB, –10.5 to +10.5V dc – 1LSB	–32,768 to +32,767
Voltage Input Impedance	210K Ω
Current Input Impedance	160Ω
Input Resolution①	16 bit
Non-linearity	< 0.002%
Overall Accuracy 0°C to +55°C	±0.7% of full scale
Overall Accuracy Drift 0°C to +55°C (max.)	±0.176%
Overall Accuracy at +25°C (+77°F) (max.)	±0.525%
Voltage Input Overvoltage Protection	24V dc
Current Input Overcurrent Protection	±50 mA
Input to Output Isolation	30V rated working/500V test 60 Hz/1s
Field Wiring to Logic Isolation	

① The analog input update rate and input resolution are a function of the input filter selection. For additional information, see page 5–3.

Analog Output Specifications

Description	Specification
Voltage Output Range	0 to 10V dc –1LSB
Current Output Range	4 to 20 mA – 1LSB
Type of Data	16-bit signed integer
Non-linearity	0.02%
Step Response	2.5 ms (at 95%)
Load Range – Voltage Output	1K Ω to ∞Ω
Load Range – Current Output	0 to 500 Ω
Output Coding 4 to 20 mA – 1 LSB, 0 to 10Vdc – 1LSB	0 to 32,767
Voltage Output Miswiring	can withstand short circuit
Current Output Miswiring	can withstand short circuit
Output Resolution	15 bit
Analog Output Settling Time	3 msec (maximum)
Overall Accuracy 0°C to +55°C	±1.0% of full scale
Overall Accuracy Drift 0°C to +55°C (max.)	±0.28%
Overall Accuracy at +25°C (+77°F) (max.) – Current Output	0.2%
Field Wiring to Logic Isolation	30V rated working/500V isolation

Input Filter Response Times (Discrete)

The input filter response time is the time from when the external input voltage reaches an on or off state to when the micro controller recognizes that change of state. The higher you set the response time, the longer it takes for the input state change to reach the micro controller. However, setting higher response times also provides better filtering of high frequency noise.

You can apply a unique input filter setting to each of the three input groups:

- 0 and 1
- 2 and 3
- 4 to x; where x=9 for 16 I/O point controllers, and x=19 for 32 I/O point controllers

The minimum and maximum response times associated with each input filter setting can be found in the tables that follow.

Response Times for High-Speed dc Inputs 0 to 3 (applies to 1761-L10BWA, 1761-L16BWA, -L20BWA-5A, -L32BWA, -L10BWB, -L16BWB, -L20BWB-5A, -L32BWB, -L16BBB, and -L32BBB controllers)

Maximum High-Speed Counter Frequency @ 50% Duty Cycle (Khz)	Nominal Filter Setting (ms)	Maximum ON Delay (ms)	Maximum OFF Delay (ms)
6.600	0.075	0.075	0.075
5.000	0.100	0.100	0.100
2.000	0.250	0.250	0.250
1.000	0.500	0.500	0.500
0.500	1.000	1.000	1.000
0.200	2.000	2.000	2.000
0.125	4.000	4.000	4.000
0.062	8.000①	8.000	8.000
0.031	16.000	16.000	16.000

① This is the default setting.

Response Times for dc Inputs 4 and Above (applies to 1761-L10BWA, 1761-L16BWA, -L20BWA-5A, -L32BWA, -L10BWB, -L16BWB, -L20BWB-5A, -L32BWB, -L16BBB, and -L32BBB controllers)

Nominal Filter Setting (ms)	Maximum ON Delay (ms)	Maximum OFF Delay (ms)
0.50	0.500	0.500
1.00	1.00	1.000
2.00	2.000	2.000
4.00	4.000	4.000
8.00①	8.000	8.000
16.00	16.000	16.000

① This is the default setting.

Response Times for ac Inputs (applies to 1761-L16AWA, -L20AWA-5A, -L32AWA, and -L32AAA controllers)

Nominal Filter Setting (ms)①	Maximum ON Delay (ms)	Maximum OFF Delay (ms)
8.0	20.0	20.0

① There is only one filter setting available for the ac inputs. If you make another selection the controller changes it to the ac setting and sets the input filter modified bit (S:5/13).

Controller Dimensions

Refer to the following table for the controller dimensions.

<table>
<tr><th>Controller: 1761-</th><th>Length: mm (in.)</th><th>Depth: mm (in.)①</th><th>Height: mm (in.)</th></tr>
<tr><td>L10BWA</td><td>120 (4.72)</td><td rowspan="8">73 (2.87)</td><td rowspan="15">80 (3.15)</td></tr>
<tr><td>L16AWA</td><td>133 (5.24)</td></tr>
<tr><td>L16BWA</td><td>120 (4.72)</td></tr>
<tr><td>L20AWA-5A</td><td rowspan="5">200 (7.87)</td></tr>
<tr><td>L20BWA-5A</td></tr>
<tr><td>L32AWA</td></tr>
<tr><td>L32BWA</td></tr>
<tr><td>L32AAA</td></tr>
<tr><td>L10BWB</td><td rowspan="3">120 (4.72)</td><td rowspan="6">40 (1.57)</td></tr>
<tr><td>L16BBB</td></tr>
<tr><td>L16BWB</td></tr>
<tr><td>L20BWB-5A</td><td rowspan="3">200 (7.87)</td></tr>
<tr><td>L32BBB</td></tr>
<tr><td>L32BWB</td></tr>
</table>

① Add approximately 13 mm (0.51 in.) when using the 1761-CBL-PM02 or 1761-CBL-HM02 communication cables.

For a template to help you install your controller, see the *MicroLogix 1000 Programmable Controllers Installation Instructions*, publication 1761-5.1.2 or the *MicroLogix 1000 (Analog) Programmable Controllers Installation Instructions*, publication 1761-5.1.3 that were shipped with your controller.

Replacement Parts

Description	Catalog Number
10 pt. ac input, 6 pt. relay output, ac power supply controller	1761-L16AWA
12 pt. ac and 4 pt. analog inputs, 8 pt. relay and 1 pt. analog outputs, ac power supply controller	1761-L20AWA-5A
20 pt. ac input, 12 pt. relay output, ac power supply controller	1761-L32AWA
6 pt. dc input, 4 pt. relay output, ac power supply controller	1761-L10BWA
10 pt. dc input, 6 pt. relay output, ac power supply controller	1761-L16BWA
12 pt. dc and 4 pt. analog inputs, 8 pt. relay and 1 pt. analog outputs, ac power supply controller	1761-L20BWA-5A
20 pt. dc input, 12 pt. relay output, ac power supply controller	1761-L32BWA
6 pt. dc input, 4 pt. relay output, dc power supply controller	1761-L10BWB
10 pt. dc input, 6 pt. relay output, dc power supply controller	1761-L16BWB
12 pt. dc and 4 pt. analog inputs, 8 pt. relay and 1 pt. analog outputs, dc power supply controller	1761-L20BWB-5A
20 pt. dc input, 12 pt. relay output, dc power supply controller	1761-L32BWB
10 pt. dc input, 4 pt. FET and 2 pt. relay outputs, dc power supply controller	1761-L16BBB
20 pt. dc input, 10 pt. FET and 2 pt. relay outputs, dc power supply controller	1761-L32BBB
20 pt. ac input, 10 pt. triac and 2 pt. relay outputs, ac power supply controller	1761-L32AAA
Terminal doors for -L16AWA (2 doors per package)	1761-RPL-T16A
Terminal doors for -L16BWA (2 doors per package)	1761-RPL-T16B
Terminal doors for -L32AWA, -L32BWA, or -L32AAA (2 doors per package)	1761-RPL-T32X
Communications door (1 door per package)	1761-RPL-COM
DIN rail latches (2 per package)	1761-RPL-DIN
2.00 m (6.56 ft) cable (DIN-to-DIN) for use with the MicroLogix 1000 HHP	1761-CBL-HM02
Hand-Held Programmer (includes 1761-CBL-HM02 communication cable)	1761-HHP-B30
Memory module for 1761-HHP-B30 (stores 1 program)	1761-HHM-K08
Memory module for 1761-HHP-B30 (stores 8 programs)	1761-HHM-K64
Memory module door for 1761-HHP-B30 (1 door per package)	1761-RPL-TRM

B Programming Reference

This appendix lists the:

- controller status file
- instruction execution times and instruction memory usage

Controller Status File

The status file lets you monitor how your operating system works and lets you direct how you want it to work. This is done by using the status file to set up control bits and monitor both hardware and programming device faults and other status information.

Note *Do not write to reserved words in the status file. If you intend writing to status file data, it is imperative that you first understand the function fully.*

The status file S: contains the following words:

Word	Function	Page
S:0	Arithmetic Flags	B–3
S:1L (low byte)	Controller Mode Status/Control (low)	B–5
S:1H (high byte)	Controller Mode Status/Control (Hi)	B–5
S:2L (low byte)	Controller Alternate Mode Status/Control (low)	B–8
S:2H (high byte)	Controller Alternate Mode Status/Control (Hi)	B–8
S:3L (low byte)	Current Scan Time	B–11
S:3H (high byte)	Watchdog Scan Time	B–11
S:4	Timebase	B–12
S:5	Minor Error Bits	B–12
S:6	Major Error Code	B–14
S:7	Suspend Code	B–18
S:8 to S:12	Reserved	B–18
S:13, S:14	Math Register	B–18
S:15L (low byte)	DF1 Full or Half-Duplex Node Address	B–18
S:15H (high byte)	DF1 Full or Half-Duplex Baud Rate	B–19
S:16L (low byte)	DH-485 Node Address	B–19
S:16H (high byte)	DH-485 Baud Rate	B–19
S:17 to S:21	Reserved	B–19
S:22	Maximum Observed Scan Time	B–19
S:23	Reserved	B–20
S:24	Index Register	B–20
S:25 to S:29	Reserved	B–20
S:30	STI Setpoint	B–20
S:31 and S:32	Reserved	B–20

Status File Descriptions

The following tables describe the status file functions, beginning at address S:0 and ending at address S:32.

Each status bit is classified as one of the following:

- Status — Use these words, bytes, or bits to monitor controller operation or controller status information. The information is seldom written to by the user program or programming device (unless you want to reset or clear a function such as a monitor bit).
- Dynamic Configuration — Use these words, bytes, or bits to select controller options while online with the controller.
- Static Configuration — Use these words, bytes, or bits to select controller options while in the offline program mode, prior to downloading the user program.

Address	Bit	Classification	Description
S:0	Arithmetic and Scan Status Flags		The arithmetic flags are assessed by the controller following the execution of certain math and data handling instructions. The state of these bits remain in effect until certain math or data handling instructions in the program are executed.
S:0/0	Carry	Status	This bit is set by the controller if a mathematical carry or borrow is generated. Otherwise the bit remains cleared. This bit is assessed as if a function of unsigned math. When a STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of S:0/0 is restored when execution resumes.
S:0/1	Overflow	Status	This bit is set by the controller when the result of a mathematical operation does not fit in its destination. Otherwise the bit remains cleared. Whenever this bit is set, the overflow trap bit S:5/0 is also set except for the ENC bit. Refer to S:5/0. When a STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of S:0/1 is restored when execution resumes.

Address	Bit	Classification	Description
S:0/2	Zero	Status	This bit is set by the controller when the result of certain math or data handling instructions is zero. Otherwise the bit remains cleared. When a STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of S:0/2 is restored when execution resumes.
S:0/3	Sign	Status	This bit is set by the controller when the result of certain math or data handling instructions is negative. Otherwise the bit remains cleared. When a STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of S:0/3 is restored when execution resumes.
S:0/4 to S:0/7	Reserved	NA	NA
S:0/8①	Extend I/O Configuration	Static Configuration	This bit must be set by the user when unused outputs are written to. If reset and unused outputs are turned on the controller will fault (41H).
S:0/9	Reserved	NA	NA
S:0/10	Primary Protocol	Static Configuration	This bit defines the protocol that the controller will initially use when attempting to establish communication, where: 0 = DF1 (default setting) 1 = DH-485
S:0/11	Active Protocol	Status	This bit is updated by the controller during a protocol switch. It indicates which protocol is currently being used for communication, where: 0 = DF1 1 = DH-485
S:0/12	Selected DF1 Protocol	Status	This bit allows the user to determine which DF1 protocol is configured, where: 0 = DF1 Full-Duplex (default setting) 1 = DF1 Half-Duplex Slave
S:0/13 to S:0/15	Reserved	NA	NA

① Valid for Series A–C discrete only.

Address	Bit	Classification	Description
S:1/0 to S:1/4	Controller Mode Status/ Control	Status	Bits 0–4 function as follows: 0 0000 = (0) Remote Download in progress 0 0001 = (1) Remote Program mode 0 0011 = (3) Suspend Idle (operation halted by SUS instruction execution) 0 0110 = (6) Remote Run mode 0 0111 = (7) Remote Test continuous mode 0 1000 = (8) Remote Test single scan mode
S:1/5	Forces Enabled	Status	This bit is set by the controller (1) to indicate that forces are always enabled.
S:1/6	Forces Installed	Status	This bit is set by the controller to indicate that forces have been set by the user.
S:1/7	Comms Active	Status	This bit is set when the controller receives valid data from the communication port. For DF1 protocols, the bit is reset if the controller does not receive valid data from the programming port for 10 seconds. **Note:** In DF1 half-duplex mode, simple polls by the DF1 master or replies to received messages will not reset the timer. A poll with a command is required to reset the timer. For DH-485, the bit is reset as soon as the DH-485 link layer determines that no other devices are active on the link. **Application Note:** For DF1 half-duplex, you can use this bit to enable a timer (via an XIO instruction) to sense whether the DF1 master is actively communicating to the slave. The preset of the timer is determined by the total network timing.
S:1/8	Fault Override at Powerup	Static Configuration	When set, this bit causes the controller to clear the Major Error Halted bit S:1/13 and Minor error bits S:5/0 to S:5/7 on power up if the processor had previously been in the REM Run mode and had faulted. The controller then attempts to enter the REM Run mode. Set this bit offline only.

Address	Bit	Classification	Description
S:1/9	Startup Protection Fault	Static Configuration	When this bit is set and power is cycled while the controller is in the REM Run mode, the controller executes the user-fault routine prior to the execution of the first scan of your program. You have the option of clearing the Major Error Halted bit S:1/13 to resume operation in the REM Run mode. If the user-fault routine does not reset bit S:1/13, the fault mode results. Program the user-fault routine logic accordingly. When executing the startup protection fault routine, S:6 (major error fault code) will contain the value 0016H.
S:1/10 to S:1/11	Reserved	NA	NA
S:1/12	Run Always	Static Configuration	When set, this bit causes the controller to clear S:1/13 before attempting to enter RUN mode when power is applied or if an unexpected reset occurs. If this bit is not set, the controller powers up in the previous mode it was in before losing power, unless the controller was in REM test mode. If the controller was in REM test mode when power was removed, the controller enters REM program mode when power is applied. This bit overrides any faults existing at power down. **Setting the Run Always bit causes the controller to enter the REM Run mode if an unexpected reset occurs, regardless of the mode that the controller is in before the reset occurred. Unexpected resets may occur due to electromagnetic noise, improper grounding, or an internal controller hardware failure. Make sure your application is designed to safely handle this situation.**

Address	Bit	Classification	Description
S:1/13	Major Error Halted	Dynamic Configuration	This bit is set by the controller any time a major error is encountered. The controller enters a fault condition. Word S:6, the Fault Code will contain a code that can be used to diagnose the fault condition. Any time bit S:1/13 is set, the controller: • either places all outputs in a safe state (outputs are off) and energizes the fault LED, • or enters the user-fault routine with outputs active (if in REM Run mode), allowing the fault routine ladder logic to attempt recovery from the fault condition. If the user-fault routine determines that recovery is required, clear S:1/13 using ladder logic prior to exiting the fault routine. If the fault routine ladder logic does not understand the fault code, or if the routine determines that it is not desirable to continue operation, the controller exits the fault routine with bit S:1/13 set. The outputs are placed in a safe state and the FAULT LED is energized. When you clear bit S:1/13 using a programming device, the controller mode changes from fault to Remote Program. You can move a value to S:6, then set S:1/13 in your ladder program to generate an application specific major error. All application generated faults are recoverable regardless of the value used. **Note:** *Once a major fault state exists, you must correct the condition causing the fault, and you must also clear this bit in order for the controller to accept a mode change attempt (into REM Run or REM Test). Also, clear S:6 to avoid the confusion of having an error code but no fault condition.* **Note:** *Do not re-use error codes that are defined later in this appendix as application specific error codes. Instead, create your own unique codes. This prevents you from confusing application errors with system errors. We recommend using error codes FFOO to FFOF to indicate application specific major errors.*

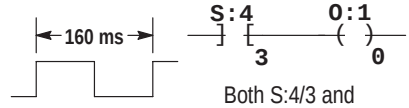
Address	Bit	Classification	Description
S:1/14	OEM Lock	Static Configuration	Using this bit you can control access to a controller file. To program this feature, select "Future Access Disallow" when saving your program. When this bit is cleared, it indicates that any compatible programming device can access the ladder program (provided that password conditions are satisfied).
S:1/15	First Pass	Status	Use this bit to initialize your program as the application requires. When this bit is set by the controller, it indicates that the first scan of the user program is in progress (following power up in the RUN mode or entry into a REM Run or REM Test mode). The controller clears this bit following the first scan. This bit is set during execution of the startup protection fault routine. Refer to S:1/9 for more information.
S:2/0	STI Pending	Status	When set, this bit indicates that the STI timer has timed out and the STI routine is waiting to be executed. This bit is cleared upon starting the STI routine, ladder program, exit of the REM Run or Test mode, or execution of a true STS instruction.
S:2/1	STI Enabled	Status and Static Configuration	This bit may be set or reset using the STS, STE, or STD instruction. If set, it allows execution of the STI if the STI setpoint S:30 is non-zero. If clear, when an interrupt occurs, the STI subroutine does not execute and the STI Pending bit is set. The STI Timer continues to run when this bit is disabled. The STD instruction clears this bit. If this bit is set or reset editing the status file online, the STI is not affected. If this bit is set, the bit allows execution of the STI. If this bit is reset editing the status file offline, the bit disallows execution of the STI.
S:2/2	STI Executing	Status	When set, this bit indicates that the STI timer has timed out and the STI subroutine is currently being executed. This bit is cleared upon completion of the STI routine, ladder program, or REM Run or Test mode.
S:2/3 to S:2/4	Reserved	NA	NA

Address	Bit	Classification	Description
S:2/5①	Incoming Command Pending Bit	Status	This bit is set when the processor determines that another node on the network has requested information or supplied a command to it. This bit can be set at any time. This bit is cleared when the processor services the request (or command).
S:2/6①	Message Reply Pending Bit	Status	This bit is set when another node on the network has supplied the information you requested in the MSG instruction of your processor. This bit is cleared when the processor stores the information and updates your MSG instruction.
S:2/7①	Outgoing Message Command Pending Bit	Status	This bit is set when one or more messages in your program are enabled and waiting, but no message is being transmitted at the time. As soon as transmission of a message begins, the bit is cleared. After transmission, the bit is set again if there are further messages waiting. It remains cleared if there are no further messages waiting.
S:2/8 to S:2/13	Reserved	NA	NA

① Valid for Series C discrete only.

Address	Bit	Classification	Description
S:2/14	Math Overflow Selection	Dynamic Configuration	Set this bit when you intend to use 32-bit addition and subtraction. When S:2/14 is set, and the result of an ADD, SUB, MUL, or DIV instruction cannot be represented in the destination address (underflow or overflow), • the overflow bit S:0/1 is set, • the overflow trap bit S:5/0 is set, • and the destination address contains the unsigned truncated least significant 16 bits of the result. The default condition of S:2/14 is reset (0). When S:2/14 is reset, and the result of an ADD, SUB, MUL, or DIV instruction cannot be represented in the destination address (underflow or overflow), • the overflow bit S:0/1 is set, • the overflow trap bit S:5/0 is set, • and the destination address contains 32767 if the result is positive or – 32768 if the result is negative. Note, the status of bit S:2/14 has no effect on the DDV instruction. Also, it has no effect on the math register content when using MUL and DIV instructions. To provide protection from inadvertent alteration of your selection, program an unconditional OTL instruction at address S:2/14 to ensure the new math overflow operation. Program an unconditional OTU instruction at address S:2/14 to ensure the original math overflow operation.
S:2/15	Reserved	NA	NA

Address	Bit	Classification	Description
S:3L	Current Scan Time	Status	The value of this byte tells you how much time elapses in a program cycle. A program cycle includes: • scanning the ladder program, • housekeeping, • scanning the I/O, • servicing of the communication channel. The byte value is zeroed by the controller each scan, immediately preceding the execution of rung 0 of program file 2 (main program file). The byte is incremented every 10 ms thereafter, and indicates, in 10 ms increments, the amount of time elapsed in each scan. If this value ever equals the value in S:3H Watchdog, a user watchdog major error will be declared (code 0022). The resolution of the scan time value is +0 to 90 ms (–10 ms). Example: The value 9 indicates that 80–90 ms has elapsed since the start of the program cycle.
S:3H	Watchdog Scan Time	Dynamic Configuration	This byte value contains the number of 10 ms ticks allowed to occur during a program cycle. The default value is 10 (100 ms), but you can increase this to 255 (2.55 seconds) or decrease it to 1, as your application requires. If the program scan S:3L value equals the watchdog value, a watchdog major error will be declared (code 0022).

S:4	Timebase	Status	All 16 bits of this word are assessed by the controller. The value of this word is zeroed upon power up in the REM Run mode or entry into the REM Run or REM Test mode. It is incremented every 10 ms thereafter. Application note: You can write any value to S:4. It will begin incrementing from this value. You can use any individual bit of this word in your user program as a 50% duty cycle clock bit. Clock rates for S:4/0 to S:4/15 are: 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480, 40960, 81920, 163840, 327680, and 655360 ms. The application using the bit must be evaluated at a rate more than two times faster than the clock rate of the bit. In the example below, bit S:4/3 toggles every 80 ms, producing a 160 ms clock rate. To maintain accuracy of this bit in your application, the instruction using bit S:4/3 (O:1/0 in this case) must be evaluated at least once every 79.999 ms  S:4/3 cycles in 160 ms Both S:4/3 and Output O:1/0 toggle every 80 ms. O:1/0 must be evaluated at least once every 79.999 ms.
S:5	Minor Error Bits		The bits of this word are set by the controller to indicate that a minor error has occurred in your ladder program. Minor errors, bits 0 to 7, revert to major error 0020H if any bit is detected as being set at the end of the scan. These bits are automatically cleared on a power cycle.
S:5/0	Overflow Trap	Dynamic Configuration	When this bit is set by the controller, it indicates that a mathematical overflow has occurred in the ladder program. See S:0/1 for more information. If this bit is ever set upon execution of the END or TND instruction, major error (0020) is declared. To avoid this type of major error from occurring, examine the state of this bit following a math instruction (ADD, SUB, MUL, DIV, DDV, NEG, SCL, TOD, or FRD), take appropriate action, and then clear bit S:5/0 using an OTU instruction with S:5/0.

S:5/1	Reserved	NA	NA
S:5/2	Control Register Error	Dynamic Configuration	The LFU, LFL, FFU, FFL, BSL, BSR, SQO, SQC, and SQL instructions are capable of generating this error. When bit S:5/2 is set, it indicates that the error bit of a control word used by the instruction has been set. If this bit is ever set upon execution of the END or TND instruction, major error (0020) is declared. To avoid this type of major error from occurring, examine the state of this bit following a control register instruction, take appropriate action, and then clear bit S:5/2 using an OTU instruction with S:5/2.
S:5/3	Major Error Detected While Executing user-fault routine	Dynamic Configuration	When set, the major error code (S:6) represents the major error that occurred while processing the fault routine due to another major error.
S:5/4 to S:5/7	Reserved	NA	NA
S:5/8	Retentive Data Lost	Status	This bit is set whenever retentive data is lost. This bit remains set until you clear it. While set, this bit causes the controller to fault prior to the first true scan of the program.
S:5/9	Reserved	NA	NA
S:5/10	STI Lost	Status	This bit is set whenever the STI timer expires while the STI routine is either executing or disabled *and* the pending bit (s:2/0) is already set.
S:5/11 to S:5/12	Reserved	NA	NA
S:5/13	Input Filter Selection Modified	Status	This bit is set whenever the discrete input filter selection in the controller is made compatible with the hardware. Refer to page A–7 for more information.
S:5/14 to S:5/15	Reserved	NA	NA

NA = Not Applicable

S:6	Major Error Code	Status	A hexadecimal code is entered in this word by the controller when a major error is declared. Refer to S:1/13. The code defines the type of fault, as indicated on the following pages. This word is not cleared by the controller. Error codes are presented, stored, and displayed in a hexadecimal format. If you enter a fault code as a parameter in an instruction in your ladder program, you must convert the code to decimal. **Application note**: You can declare your own application specific major fault by writing a unique value to S:6 and then setting bit S:1/13. Interrogate the value of S:6 in the user-fault routine to determine the type of fault that occurred. Fault Classifications: Faults are classified as Non-User, Non-Recoverable, and Recoverable. Error code descriptions and classifications are listed on the following pages. Categories are: • powerup errors • going-to-run errors • run errors • download errors

NA = Not Applicable

Each fault is classified as one of the following:

- Non-User — A fault caused by various conditions that cease ladder program execution. The user-fault routine is not run when this fault occurs.
- Non-Recoverable — A fault caused by the user that cannot be recovered from. The user-fault routine is run when this fault occurs. However, the fault cannot be cleared.
- Recoverable — A fault caused by the user that can be recovered from in the user-fault routine by resetting major error halted bit (S:1/13). The user-fault routine is run when this fault occurs.

Refer to chapter 14, Troubleshooting, for more information regarding programming device advisory messages.

Address	Error Code (Hex)	*Powerup Errors*	Fault Classification		
			Non-User	User	
				Non-Recoverable	Recoverable
S:6	0001	The default program was loaded.	X		
	0002	Unexpected reset occurred.	X		
	0003	EEPROM memory is corrupt.	X		
	0008	A fatal internal programming device error occurred.	X		
	0009	A fatal internal hardware error occurred.	X		

Address	Error Code (Hex)	*Going-to-Run (GTR) Errors*①	Fault Classification		
			Non-User	User	
				Non-Recoverable	Recoverable
S:6	0005	Retentive data is lost.			X
	0010	The downloaded program is not a controller program.	X		
	0016	Startup protection after power loss, S:1/9 is set. The user must check for a retentive data lost condition if the user-fault routine was executed with startup protection.			X

① Going-to-Run errors occur when the controller is going from any mode to REM Run mode or from any non-Run mode (PRG, SUS) to Test mode.

Address	Error Code (Hex)	*Run Errors*	Fault Classification		
			Non-User	User	
				Non-Recoverable	Recoverable
S:6	0004	A runtime memory integrity error occurred.	X		
	0020	A minor error at the end of the scan. Refer to S:5.			X
	0022	The watchdog timer expired. Refer to S:3H.		X	
	0024	Invalid STI interrupt setpoint. Refer to S:30.		X	
	0025	There are excessive JSRs in the STI subroutine (file 5).		X	
	0027	There are excessive JSRs in the fault subroutine (file 3).		X	
	002A	The indexed address is too large for the file.		X	
	002B	There are excessive JSRs in the high-speed counter sub-routine (file 4).		X	
	0030	The subroutine nesting exceeds a limit of 8 (file 2).		X	
	0031	An unsupported instruction was detected.	X		
	0032	An SQO/SQC instruction crossed data file boundaries.			X
	0033	The LFU, LFL, FFU, FFL, BSL, or BSR instruction crossed data file boundaries.			X
	0034	A negative value for a timer accumulator or preset value was detected.			X
	0035	An illegal instruction (TND) occurred in the interrupt file.		X	
	0037	Invalid presets were loaded to the high-speed counter.			X
	0038	A RET instruction was detected in program file 2.	X		

Address	Error Code (Hex)	*Run Errors*	Fault Classification		
			Non-User	User	
				Non-Recoverable	Recoverable
S:6	0040	An output verify write occurred.		X	
	0041[1]	Extra output bit(s) turned on.		X	

[1] Valid for Series A–C discrete only.

Address	Error Code (Hex)	*Download Errors*	Fault Classification		
			Non-User	User	
				Non-Recoverable	Recoverable
S:6	0018	The user program is incompatible with the operating system.	X		

Address	Bit	Classification	Description
S:7	Suspend Code	Status	When a non-zero value appears in S:7, it indicates that the SUS instruction identified by this value has been evaluated as true, and the Suspend Idle mode is in effect. This pinpoints the conditions in the application that caused the Suspend Idle mode. This value is not cleared by the controller. Use the SUS instruction with startup troubleshooting, or as runtime diagnostics for detection of system errors.
S:8 to S:12	Reserved	NA	NA
S:13 and S:14	Math Register	Status	Use this double register to produce 32-bit signed divide and multiply operations, precision divide or double divide operations, and 5-digit BCD conversions. These two words are used in conjunction with the MUL, DIV, DDV, FRD, and TOD math instructions. The math register value is assessed upon execution of the instruction and remains valid until the next MUL, DIV, DDV, FRD, or TOD instruction is executed in the user program. An explanation of how the math register operates is included with the instruction definitions. If you store 32-bit signed data values, you must manage this data type without the aid of an assigned 32-bit data type. For example, combine B3:0 and B3:1 to create a 32-bit signed data value. We recommend that you start all 32-bit values on an even or odd word boundary for ease of application and viewing. Also, we recommend that you design, document, and view the contents of 32-bit signed data in either the hexadecimal or binary radix. When an STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of the math register is restored when execution resumes.
S:15L	DF1 Node Address	Status	This byte value contains the node address of your processor on the DF1 link. It is used when executing Message (MSG) instructions over the DF1 link. The default node address of a processor is 1. Valid node addresses are 0–254. To change a processor node address you must use a programming device.

Address	Bit	Classification	Description
S:15H	DF1 Baud Rate	Status	This byte value contains a code used to select the baud rate of the processor on the DF1 link. The controller baud rate options are: • 300 • 600 • 1200 • 2400 • 4800 • 9600 (default) • 19200 • 38400 To change the baud rate from the default value you must use a programming device.
S:16L	DH-485 Node Address	Status	This byte value contains the node address of your processor on the DH-485 link. Each device on the DH-485 link must have a unique address between the decimal values 1–31. To change a processor node address, you must use a programming device.
S:16H	DH-485 Baud Rate	Status	This byte value contains the baud rate of the processor on the DH-485 link. The controller baud rate options are: • 9600 • 19200 (default) To change the baud rate from the default value, you must use a programming device.
S:17 to S:21	Reserved	NA	NA
S:22	Maximum Observed Scan Time	Dynamic Configuration	This word indicates the maximum observed interval between consecutive program cycles. This value indicates, in 10 ms increments, the time elapsed in the longest program cycle of the controller. Refer to S:3L for more information regarding the program cycle. The controller compares each last scan value to the value contained in S:22. If the controller determines that the last scan value is larger than the value stored at S:22, the last scan value is written to S:22. Resolution of the maximum observed scan time value is +0 to –10 ms. For example, the value 9 indicates that 80–90 ms was observed as the longest program cycle. Interrogate this value if you need to determine or verify the longest scan time of your program.

Address	Bit	Classification	Description
S:23	Reserved	NA	NA
S:24	Index Register	Status	This word indicates the element offset used in indexed addressing. When an STI, high-speed counter, or Fault Routine interrupts normal execution of your program, the original value of this register is restored when execution resumes.
S:25 to S:29	Reserved	NA	NA
S:30	STI Setpoint	Dynamic Configuration	You enter the timebase to be used in the selectable timed interrupt (STI). The time can range from 10 to 2550 ms. (This is in 10ms increments, so valid values are from 0–255.) Your STI routine executes per the value you enter. Write a zero value to disable the STI. To provide protection from inadvertent alteration of your selection, program an unconditional MOV instruction containing the setpoint value of your STI to S:30, or program a CLR instruction at S:30 to prevent STI operation. If the STI is initiated while in the REM Run mode by loading the status registers, the interrupt starts timing from the end of the program scan in which the status registers were loaded.
S:31 to S:32	Reserved	NA	NA

Instruction Execution Times and Memory Usage

The table below lists the execution times and memory usage for the controller instructions. Any instruction that takes longer than 15 µs (true or false execution time) to execute performs a poll for user interrupts.

Mnemonic	False Execution Time (approx. µseconds)	True Execution Time (approx. µseconds)	Memory Usage (user words)	Name	Instruction Type
ADD	6.78	33.09	1.50	Add	Math
AND	6.78	34.00	1.50	And	Data Handling
BSL	19.80	53.71 + 5.24 x position value	2.00	Bit Shift Left	Application Specific
BSR	19.80	53.34 + 3.98 x position value	2.00	Bit Shift Right	Application Specific
CLR	4.25	20.80	1.00	Clear	Math
COP	6.60	27.31 + 5.06/word	1.50	File Copy	Data Handling
CTD	27.22	32.19	1.00	Count Down	Basic
CTU	26.67	29.84	1.00	Count Up	Basic
DCD	6.78	27.67	1.50	Decode 4 to 1 of 16	Data Handling
DDV	6.78	157.06	1.00	Double Divide	Math
DIV	6.78	147.87	1.50	Divide	Math
ENC	6.78	54.80	1.50	Encode 1 of 16 to 4	Data Handling
EQU	6.60	21.52	1.50	Equal	Comparison
FFL	33.67	61.13	1.50	FIFO Load	Data Handling
FFU	34.90	73.78 + 4.34 x position value	1.50	FIFO Unload	Data Handling
FLL	6.60	26.86 + 3.62/word	1.50	Fill File	Data Handling
FRD	5.52	56.88	1.00	Convert from BCD	Data Handling
GEQ	6.60	23.60	1.50	Greater Than or Equal	Comparison
GRT	6.60	23.60	1.50	Greater Than	Comparison
HSC	21.00	21.00	1.00	High-Speed Counter	High-Speed Counter

Mnemonic	False Execution Time (approx. μseconds)	True Execution Time (approx. μseconds)	Memory Usage (user words)	Name	Instruction Type
HSD	7.00	8.00	1.25	High-Speed Counter Interrupt Disable	High-Speed Counter
HSE	7.00	10.00	1.25	High-Speed Counter Interrupt. Enable	High-Speed Counter
HSL	7.00	66.00	1.50	High-Speed Counter Load	High-Speed Counter
IIM	6.78	35.72	1.50	Immediate Input with Mask	Program Flow Control
INT	0.99	1.45	0.50	Interrupt Subroutine	Application Specific
IOM	6.78	41.59	1.50	Immediate Output with Mask	Program Flow Control
JMP	6.78	9.04	1.00	Jump to Label	Program Flow Control
JSR	4.25	22.24	1.00	Jump to Subroutine	Program Flow Control
LBL	0.99	1.45	0.50	Label	Program Flow Control
LEQ	6.60	23.60	1.50	Less Than or Equal	Comparison
LES	6.60	23.60	1.50	Less Than	Comparison
LIM	7.69	36.93	1.50	Limit Test	Comparison
LFL	33.67	61.13	1.50	LIFO Load	Data Handling
LFU	35.08	64.20	1.50	LIFO Unload	Data Handling
MCR	4.07	3.98	0.50	Master Control Reset	Program Flow Control
MEQ	7.69	28.39	1.50	Masked Comparison for Equal	Comparison
MOV	6.78	25.05	1.50	Move	Data Handling

Mnemonic	False Execution Time (approx. μseconds)	True Execution Time (approx. μseconds)	Memory Usage (user words)	Name	Instruction Type
MSG	26	180①②	34.75	Message	Communication
MUL	6.78	57.96	1.50	Multiply	Math
MVM	6.78	33.28	1.50	Masked Move	Data Handling
NEG	6.78	29.48	1.50	Negate	Data Handling
NEQ	6.60	21.52	1.50	Not Equal	Comparison
NOT	6.78	28.21	1.00	Not	Data Handling
OR	6.78	33.68	1.50	Or	Data Handling
OSR	11.48	13.02	1.00	One-Shot Rising	Basic
OTE	4.43	4.43	0.75	Output Energize	Basic
OTE (high-speed counter)	7.00	12.00	0.75	Update High-Speed Counter Image Accumulator	High-Speed Counter
OTL	3.16	4.97	0.75	Output Latch	Basic
OTU	3.16	4.97	0.75	Output Unlatch	Basic
RAC	6.00	56.00	1.00	High-Speed Counter Reset Accumulator	High-Speed Counter
RES (timer/counter)	4.25	15.19	1.00	Reset	Basic
RES (high-speed counter)	6.00	51.00	1.00	High-Speed Counter Reset	High-Speed Counter
RET	3.16	31.11	0.50	Return from Subroutine	Program Flow Control
RTO	27.49	38.34	1.00	Retentive Timer	Basic
SBR	0.99	1.45	0.50	Subroutine	Program Flow Control
SCL	6.78	169.18	1.75	Scale Data	Math
SQC	27.40	60.52	2.00	Sequencer Compare	Application Specific

① This only includes the amount of time needed to set up the operation requested. It does not include the time it takes to service the actual communication, as this time varies with each network configuration. As an example, 144ms is the actual communication service time for the following configuration: 3 nodes on DH-485 (2=MicroLogix 1000 programmable controllers and 1=PLC-500 A.I. Series™ programming software), running at 19.2K baud, with 2 words per transfer.

② Add 7.3 μseconds per word for MSG instructions that perform writes.

Mnemonic	False Execution Time (approx. μseconds)	True Execution Time (approx. μseconds)	Memory Usage (user words)	Name	Instruction Type
SQL	28.12	53.41	2.00	Sequencer Load	Application Specific
SQO	27.40	60.52	2.00	Sequencer Output	Application Specific
SQR	6.78	71.25	1.25	Square Root	Math
STD	3.16	6.69	0.50	Selectable Timer Interrupt Disable	Application Specific
STE	3.16	10.13	0.50	Selectable Timer Interrupt Enable	Application Specific
STS	6.78	24.59	1.25	Selectable Timer Interrupt Start	Application Specific
SUB	6.78	33.52	1.50	Subtract	Math
SUS	7.87	10.85	1.50	Suspend	Program Flow Control
TND	3.16	7.78	0.50	Temporary End	Program Flow Control
TOD	6.78	49.64	1.00	Convert to BCD	Data Handling
TOF	31.65	39.42	1.00	Timer Off-Delay	Basic
TON	30.38	38.34	1.00	Timer On-Delay	Basic
XIC	1.72	1.54	0.75	Examine If Closed	Basic
XIO	1.72	1.54	0.75	Examine If Open	Basic
XOR	6.92	33.64	1.50	Exclusive Or	Data Handling

User Interrupt Latency

The user interrupt latency is the maximum time from when an interrupt condition occurs (e.g., STI expires or HSC preset is reached) to when the user interrupt subroutine begins executing (assumes that there are no other interrupt conditions present).

If you are communicating with the controller, the maximum user interrupt latency is 872 μs. If you are *not* communicating with the controller, the maximum user interrupt latency is 838 μs.

Estimating Memory Usage for Your Control System

Use the following to calculate memory usage for your control system.

_____	**1.** Determine the total instruction words used by the instructions in your program and enter the result. Refer to the table on page B–21.
_____	**2.** Multiply the total number of rungs by 0.75 and enter the result. Do not count the END rungs in each file.
177	**3.** To account for controller overhead, use 177.
110	**4.** To account for application data, use 110.
Total Memory Usage: _____	**5.** Total steps 1 through 4. This is the estimated total memory usage of your application system. Remember, this is an estimate, actual compiled programs may differ by ±12%.
	6. To determine the estimated amount of memory remaining in the controller you have selected, do the following:
1024	Subtract the total memory usage from 1024.
Total Memory Usage (from above): – _____	
Total Memory Remaining: _____	The result of this calculation will be the estimated total memory remaining in your selected controller.
Note	*The calculated memory usage may vary from the actual compiled program by ±12%.*

Execution Time Worksheet

Use this worksheet to calculate your execution time for ladder program.

Procedure	Maximum Scan Time
1. Input scan time, output scan time, housekeeping time, and forcing.	210 µs (discrete) 330 µs with forcing (analog) 250 µs without forcing (analog)
2. Estimate your program scan time:	
A. Count the number of program rungs in your logic program and multiply by 6.	_____
B. Add up your program execution times when all instructions are true. Include interrupt routines in this calculation.①	_____ µs
3. Estimate your controller scan time:	
A. Without communications, add sections 1 and 2	_____ µs
B. With communications, add sections 1 and 2 and multiply by 1.05	_____ µs
4. To determine your maximum scan time in ms, divide your controller scan time by 1000.	_____ ms

① If a subroutine executes more than once per scan, include each subroutine execution scan time.

C Valid Addressing Modes and File Types for Instruction Parameters

This appendix lists all of the available programming instructions along with their parameters, valid addressing modes, and file types.

Available File Types

The following file types are available:

- O Output
- I Input
- S Status
- B Binary
- T Timer
- C Counter
- R Control
- N Integer

All file types are word addresses, unless otherwise specified.

Available Addressing Modes

The following addressing modes are available:

- immediate
- direct
- indirect

Immediate Addressing

Indicates that a constant is a valid file type.

Direct Addressing

The data stored in the specified address is used in the instruction. For example:

N7:0 T4:8.ACC
ST20:5

Indexed Direct Addressing

You may specify an address as being indexed by placing the "#" character in front of the address. When an address of this form is encountered in the program, the processor takes the element number of the address and adds to it the value contained in the Index Register S:24, then uses the result as the actual address. For example:

#N7:10 where S:24 = 15
The actual address used by the instruction is N7:25.

Instruction	Description	Instruction Parameters	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
ADD	Add	source A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
AND	Logical AND	source A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
BSL	Bit Shift Left	file	indexed direct	O, I, S, B, N	Not Applicable
		control	direct	R (element level)	Not Applicable
		bit address	direct	O, I, S, B, T, C, R, N (bit level)	Not Applicable
		length	(contained in the control register)		0–2048
BSR	Bit Shift Right	file	indexed direct	O, I, S, B, N	Not Applicable
		control	direct	R (element level)	Not Applicable
		bit address	direct	O, I, S, B, T, C, R, N (bit level)	Not Applicable
		length	(contained in the control register)		0–2048
CLR	Clear	destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
COP	Copy File	source	indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		length	immediate		1–128; 1–42 when destination is T,C,R

Instruction	Description	Instruction Parameters	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
CTD	Count Down	counter	direct	C (element level)	Not Applicable
		preset	(contained in the counter register)		–32,768–32,767
		accum	(contained in the counter register)		–32,768–32,767
CTU	Count Up	counter	direct	C (element level)	Not Applicable
		preset	(contained in the counter register)		–32,768–32,767
		accum	(contained in the counter register)		–32,768–32,767
DCD	Decode 4 to 1 of 16	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
DDV	Double Divide	source	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
DIV	Divide	source A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
ENC	Encode 1 of 16 to 4	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
EQU	Equal	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max

Instruction	Description	Instruction Parameters	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
FFL	FIFO Load	source	direct, indexed direct①	O, I, S, B, T, C, R, N	–32,768–32,767
		FIFO array	indexed direct	O, I, S, B, N	Not Applicable
		FIFO control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–128
		position	(contained in the control register)		0–127
FFU	FIFO Unload	FIFO array	indexed direct	O, I, S, B, N	Not Applicable
		destination	direct, indexed direct①	O, I, S, B, T, C, R, N	Not Applicable
		FIFO control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–128
		position	(contained in the control register)		0–127
FLL	Fill File	source	direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	indexed direct	O, I, S, B, T, C, R, N (element level)	Not Applicable
		length	immediate		1–128; 1–42 when destination is T,C,R
FRD	Convert from BCD	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
GEQ	Greater Than or Equal	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
GRT	Greater Than	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max

① Indexed addressing is not allowed when using T, C, or R addresses.

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
HSC	High-Speed Counter	type	immediate		0–7, where: 0=up 1=up&reset/hold 2=pulse/direction 3=pule/direction & reset/hold 4=up/down 5=up/down & reset/hold 6=encoder 7=encoder & reset/hold
		counter	direct	C5:0. C5:1 (element level)	Not Applicable
		preset	(contained in the counter register)		–32,768–32,767
		accum	(contained in the counter register)		–32,768–32,767
HSD	HSC Interrupt Disable	counter	direct	C	Not Applicable
HSE	HSC Interrupt Enable	counter	direct	C	Not Applicable
HSL	HSC Load	counter	direct	C	Not Applicable
		source	direct	B, N	Not Applicable
		length			always 5
IIM	Immediate Input with Mask	slot	direct	I	Not Applicable
		mask	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		length	immediate		1–10
INT	Interrupt Subroutine				Not Applicable

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
IOM	Immediate Output with Mask	slot	direct	O	Not Applicable
		mask	direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		length	immediate		1–32
JMP	Jump	label number	immediate		0–999
JSR	Jump to Subroutine	subroutine file number	immediate		3–255
LBL	Label	label number	immediate		0–999
LEQ	Less Than or Equal To	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
LES	Less Than	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
LFL	LIFO Load	source	immediate, direct, indexed direct①	O, I, S, B, T, C, R, N①	–32,768–32,767
		LIFO array	indexed direct	O, I, S, B, N	Not Applicable
		LIFO control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–128
		position	(contained in the control register)		0–127
LFU	LIFO Unload	LIFO array	indexed direct	O, I, S, B, N	Not Applicable
		destination	direct, indexed direct①	O, I, S, B, T, C, R, N	Not Applicable
		LIFO control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–128
		position	(contained in the control register)		0–127

① Indexed addressing is not allowed when using T, C, or R addresses.

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
LIM	Limit Test	low limit	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		test	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		high limit	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
MCR	Master Control Reset				Not Applicable
MEQ	Mask Comparison for Equal	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source mask	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		compare	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
MOV	Move	source	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
MSG	Message	read/write	immediate		0=read,1=write
		target device	immediate		2=500CPU, 4=485CIF
		control block	direct	N	Not Applicable
		control block length	immediate		7
		local address	direct	O, I, S, B, T, C, R, N	Not Applicable
		target node	(contained in the control register)		0–254 for DF1; 0–31 for DH-485
		target address	direct	O, I, S, B, T, C, R, N	0–255
		message length		T, C, R	1–13
				I, O, S, B, N	1–41

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
MUL	Multiply	source A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
MVM	Masked Move	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source mask	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
NEG	Negate	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
NEQ	Not Equal	source A	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
NOT	Logical NOT	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
OR	Logical OR	source A	direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		source B	direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
OSR	One–Shot Rising	bit address	direct	O, I, S, B, T, C, R, N	Not Applicable
OTE	Output Energize	bit address	direct	O, I, S, B, T, C, R, N	Not Applicable
OTL	Output Latch	bit address	direct	O, I, S, B, T, C, R, N	Not Applicable
OTU	Output Unlatch	bit address	direct	O, I, S, B, T, C, R, N	Not Applicable
RAC	HSC Reset Accumulator	counter	direct	C	Not Applicable
		source	immediate, direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
RES	Timer/Counter Reset	structure	direct	T, C, R (element level)	Not Applicable
RES	High-Speed Counter Reset	structure	direct	T, C, R (element level)	Not Applicable
RET	Return				Not Applicable
RTO	Retentive Timer	timer	direct	T (element level)	Not Applicable
		time base	immediate		0.01 or 1.00
		preset	(contained in the timer register)		0–32,767
		accum	(contained in the timer register)		0–32,767
SBR	Subroutine				Not Applicable
SCL	Scale	source	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
		rate	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		offset	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
SQC	Sequencer Compare	file	indexed direct	O, I, S, B, N	Not Applicable
		mask	immediate, direct, indexed direct①	O, I, S, B, T, C, R, N	–32,768–32,767
		source	direct, indexed direct①	O, I, S, B, T, C, R, N	Not Applicable
		control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–255
		position	(contained in the control register)		0–255

① Indexed addressing is not allowed when using T, C, or R addresses.

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
SQL	Sequencer Load	file	indexed direct	O, I, S, B, N	Not Applicable
		source	direct, indexed direct①	O, I, S, B, T, C, R, N	–32,768–32,767
		control	direct	R (element level)	Not Applicable
		length	(contained in the control register)		1–255
		position	(contained in the control register)		0–255
SQO	Sequencer Output	file	indexed direct	O, I, S, B, N	Not Applicable
		mask	direct, indexed direct①	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct①	O, I, S, B, T, C, R, N	Not Applicable
		control	direct	R (element level)	Not Applicable
		length			1–255
		position			0–255
SQR	Square Root	source	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable
STD	Selectable Timed Disable				Not Applicable
STE	Selectable Timed Enable				Not Applicable
STS	Selectable Timed Start	file	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	always equal 5
		time	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	0–255
SUB	Subtract	source A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,76 f-min–f-max
		source B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767 f-min–f-max
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable

① Indexed addressing is not allowed when using T, C, or R addresses.

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types	Valid Value Ranges
SUS	Suspend	suspend ID	immediate,		–32,768–32,767
TND	Temporary End				Not Applicable
TOD	Convert to BCD	source	direct, indexed direct	O, I, S, B, T, C, R, N	
		destination	direct	O, I. S. B. T, C, R, N	Not Applicable
TOF	Timer Off-Delay	timer	direct	T (element level)	Not Applicable
		time base	immediate		0.01 or 1.00
		preset	(contained in the timer register)		0–32,767
		accum	(contained in the timer register)		0–32,767
TON	Timer On-Delay	timer	direct	T (element level)	Not Applicable
		time base	immediate		0.01 or 1.00
		preset	(contained in the timer register)		0–32,767
		accum	(contained in the timer register)		0–32,767
XIC	Examine if Closed	source bit	direct	O, I, S, B, T, C, R, N (bit level)	Not Applicable
XIO	Examine if Open	source bit	direct	O, I, S, B, T, C, R, N (bit level)	Not Applicable
XOR	Exclusive OR	address A	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		address B	immediate, direct, indexed direct	O, I, S, B, T, C, R, N	–32,768–32,767
		destination	direct, indexed direct	O, I, S, B, T, C, R, N	Not Applicable

Notes:

D Understanding the Communication Protocols

Use the information in this appendix to understand the differences in communication protocols. The following protocols are supported from the RS-232 communication channel:

- DF1 Full-Duplex and DF1 Half-Duplex Slave

  All MicroLogix 1000 controllers support the DF1 full-duplex protocol. Series D or later discrete and all MicroLogix 1000 analog controllers also support DF1 half-duplex slave protocol.

- DH-485

  Series C or later discrete and all MicroLogix 1000 analog controllers can communicate on DH-485 networks using an AIC+ Advanced Interface Converter.

For information about required network connecting equipment, see chapter 3, Connecting the System.

RS-232 Communication Interface

RS-232 is an Electronics Industries Association (EIA) standard that specifies the electrical, mechanical, and functional characteristics for serial binary communication. It provides you with a variety of system configuration possibilities. (RS-232 is a definition of electrical characteristics; it is *not* a protocol.)

One of the biggest benefits of the RS-232 interface is that it lets you integrate telephone and radio modems into your control system (using the appropriate DF1 protocol only; not DH-485 protocol). The distance over which you are able to communicate with certain system devices is virtually limitless.

DF1 Full-Duplex Protocol

DF1 Full-Duplex communication protocol combines data transparency (ANSI — American National Standards Institute — specification subcategory D1) and 2-way simultaneous transmission with embedded responses (subcategory F1).

The MicroLogix 1000 controllers support the DF1 Full-Duplex protocol via RS-232 connection to external devices, such as computers, the Hand-Held Programmer (catalog number 1761-HHP-B30), or other MicroLogix 1000 controllers. (For information on connecting to the Hand-Held Programmer, see its user manual, publication 1761-6.2).

DF1 Full-Duplex Operation

DF1 Full-Duplex protocol (also referred to as DF1 point-to-point protocol) is useful where RS-232 point-to-point communication is required. This type of protocol supports simultaneous transmissions between two devices in both directions. DF1 protocol controls message flow, detects and signals errors, and retries if errors are detected.

DF1 Full-Duplex Configuration Parameters

When the system mode driver is DF1 Full-Duplex, the following parameters can be changed:

Parameter	Options	Default
Baud Rate	Toggles between the communication rate of 300, 600, 1200, 2400, 4800①, 9600, 19200, and 38400①.	9600②
Node Address	Valid range is 0–254 decimal for MicroLogix 1000 Series C and later discrete and all MicroLogix 1000 analog. Not configurable for MicroLogix 1000 Series A and B discrete.	1
Parity	None	No Parity
Stop Bits	None	1
Error Detection	None	CRC
DLE NAK retries	None	N retries③
DLE ENQ retries	None	N retries③
ACK Timeout	None	1 s
Duplicate Packet Detection	None	Enabled
Control Line	None	No Handshaking
Embedded Responses	None	Enabled

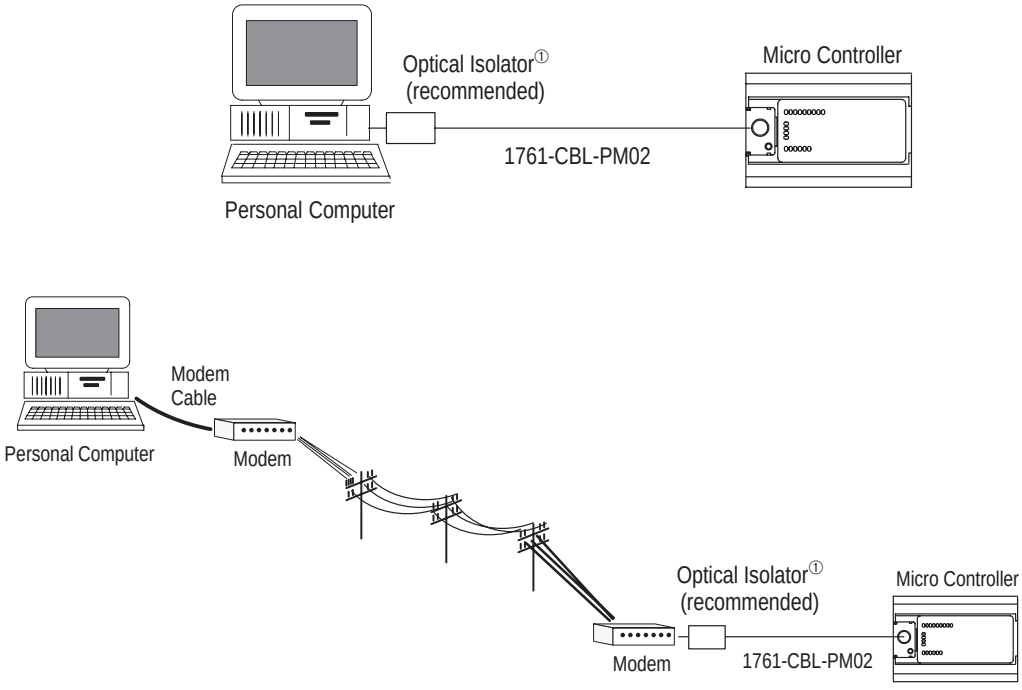
① Applicable only to MicroLogix 1000 Series D or later discrete and all MicroLogix 1000 analog controllers.

② If retentive communication data is lost, default is 1200 for MicroLogix 1000 Series A, B, or C discrete only. For MicroLogix 1000 Series D or later discrete and all MicroLogix 1000 analog, if retentive communication data is lost, baud rate defaults to 9600.

③ N=255 for MicroLogix 1000 Series A and B discrete. N=6 for MicroLogix 1000 Series C and later discrete and all MicroLogix 1000 analog.

Example DF1 Full-Duplex Connections

For information about required network connecting equipment, see chapter 3, Connecting the System.



① We recommend using an AIC+, catalog number 1761-NET-AIC, as your optical isolator. See page 3–12 for specific AIC+ cabling information.

DF1 Half-Duplex Slave Protocol

DF1 half-duplex slave protocol provides a multi-drop single master/multiple slave network. In contrast to DF1 full-duplex, communication takes place in one direction at a time. You can use the RS-232 port on the MicroLogix as both a half-duplex programming port, as well as a half-duplex peer-to-peer messaging port.

The master device initiates all communication by "polling" each slave device. The slave device may only transmit message packets when it is polled by the master. It is the master's responsibility to poll each slave on a regular and sequential basis to allow slaves to send message packets back to the master. During a polling sequence, the master polls a slave either repeatedly until the slave indicates that it has no more message packets to transmit or just one time per polling sequence, depending on how the master is configured.

An additional feature of the DF1 half-duplex protocol is that it is possible for a slave device to enable a MSG instruction in its ladder program to send or request data to/from another slave. When the initiating slave is polled, the MSG instruction command packet is sent to the master. The master recognizes that the command packet is not intended for it but for another slave, so the master immediately rebroadcasts the command packet to the intended slave. When the intended slave is polled, it sends a reply packet to the master with the data the first slave requested. The master again recognizes that the reply packet is intended for another slave, so the master immediately rebroadcasts the reply packet to that slave. This slave-to-slave transfer is a function of the master device and is also used by programming software to upload and download programs to processors on the DF1 half-duplex link.

Several Allen-Bradley products support DF1 half-duplex master protocol. They include the SLC 5/03™, SLC 5/04™ and SLC 5/05™, and enhanced PLC-5® processors. Rockwell Software WINtelligent LINX™ and RSLinx (version 2.x and higher) also support DF1 half-duplex master protocol.

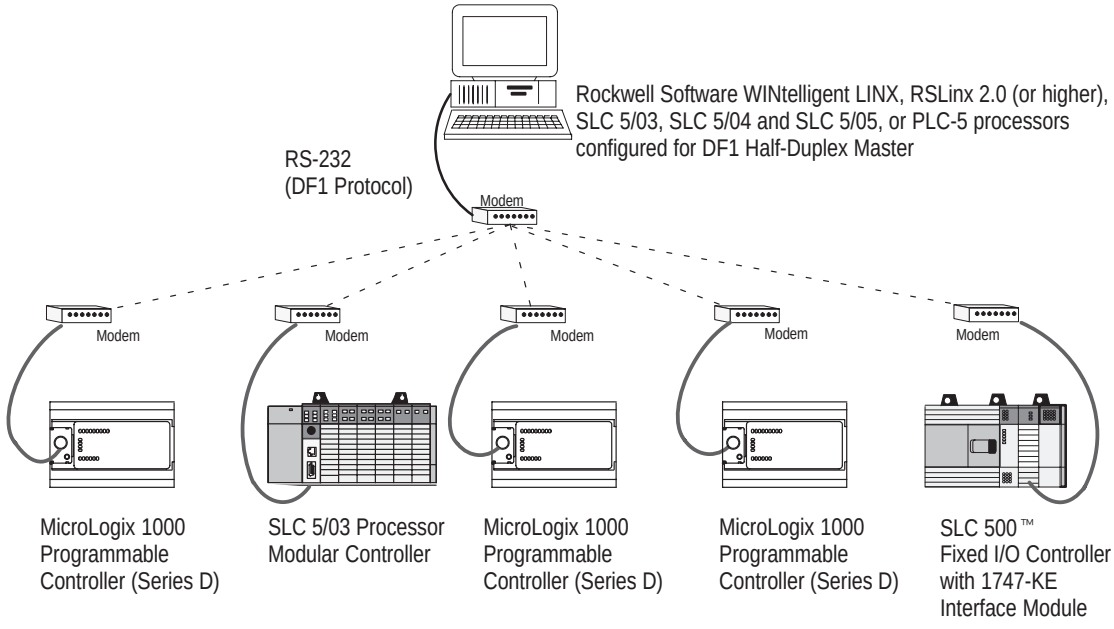
Typically, the master maintains an active node table that indicates which slaves are active (slaves that responded the last time they were polled) and which slaves are inactive (slaves that did not respond the last time they were polled). The active slaves are polled on a regular basis. The inactive slaves are only polled occasionally to check if any have come back online.

DF1 half-duplex supports up to 255 devices (address 0 to 254) with address 255 reserved for master broadcasts. The MicroLogix supports broadcast reception but cannot initiate a broadcast command. The MicroLogix supports half-duplex modems using Request-To-Send/Clear-To-Send (RTS/CTS) hardware handshaking.

DF1 Half-Duplex Slave Configuration Parameters

When the system mode driver is DF1 half-duplex slave the following parameters can be viewed and changed only when the programming software is online with the processor. The DF1 half-duplex slave parameters are not stored as part of the controller downloadable image *(with the exception of the baud rate and node address)*. If a failed MicroLogix 1000 controller is replaced and the backed-up controller image is downloaded to the replacement controller, these parameters will remain at default until manually changed. Therefore, be sure to fully document any non-default settings to the DF1 half-duplex slave configuration parameters.

Parameter	Description	Default
Baud Rate	Toggles between the communication rate of 300, 600, 1200, 2400, 4800, 9600, 19,200, and 38.4K.	9600
Node Address	Valid range is 0–254 decimal.	1
Control Line	Toggles between No Handshaking and Half-Duplex Modem.	No Hand-shaking
Duplicate Packet Detection	Detects and eliminates duplicate responses to a message. Duplicate packets may be sent under "noisy" communication conditions when the sender's retries are not set to 0. Toggles between Enabled and Disabled.	Enabled
Error Detection	Toggles between CRC and BCC.	CRC
RTS Off Delay	Specifies the delay time between when the last serial character is sent to the modem and when RTS will be deactivated. Gives modem extra time to transmit the last character of a packet. The valid range is 0–255 and can be set in increments of 5 ms.	0
RTS Send Delay	Specifies the time delay between setting RTS (request to send) until checking for the CTS (clear to send) response. For use with modems that are not ready to respond with CTS immediately upon receipt of RTS. The valid range is 0–255 and can be set in increments of 5 ms.	0
Poll Timeout	Poll Timeout only applies when a slave device initiates a MSG instruction. It is the amount of time that the slave device will wait for a poll from the master device. If the slave device does not receive a poll within the Poll Timeout, a MSG instruction error will be generated, and the ladder program will need to requeue the MSG instruction. The valid range is 0–65535 and can be set in increments of 20 ms. If you are using a MSG instruction, it is recommended that a Poll Timeout value of zero not be used. Poll Timeout is disabled if set to zero.	3000 (60s)
Pre-send Time Delay	Delay time before transmission. Required for 1761-NET-AIC physical half-duplex networks. The 1761-NET-AIC needs delay time to change from transmit to receive mode. The valid range is 0–255 and can be set in increments of 5 ms.	0
Message Retries	Specifies the number of times a slave device will attempt to resend a message packet when it does not receive an ACK from the master device. For use in noisy environments where message packets may become corrupted in transmission. The valid range is 0–255.	3
EOT Suppression	Slave does not respond when polled if no message is queued. Saves modem transmission power when there is no message to transmit. Toggles between Yes and No.	No


Rockwell Software WINtelligent LINX, RSLinx 2.0 (or higher), SLC 5/03, SLC 5/04 and SLC 5/05, or PLC-5 processors configured for DF1 Half-Duplex Master
RS-232
(DF1 Protocol)
Modem
Modem
Modem
Modem
Modem
Modem
MicroLogix 1000 Programmable Controller (Series D)
SLC 5/03 Processor Modular Controller
MicroLogix 1000 Programmable Controller (Series D)
MicroLogix 1000 Programmable Controller (Series D)
SLC 500™ Fixed I/O Controller with 1747-KE Interface Module

Considerations When Communicating as a DF1 Slave on a Multi-drop Link

When communication is between either your programming software and a MicroLogix 1000 Programmable Controller or between two MicroLogix Programmable Controllers via a slave-to-slave connection on a larger multi-drop link, the devices depend on a DF1 Master to give each of them polling permission to transmit in a timely manner. As the number of slaves increases on the link (up to 254), the time between when your programming software or the MicroLogix Controller is polled also increases. This increase in time may become larger if you are using low baud rates.

As these time periods grow, the following values may need to be changed to avoid loss of communication:

- programming software - increase poll timeout and reply timeout values
- MicroLogix Programmable Controller - increase poll timeout

Ownership Timeout

When a program download sequence is started by a software package to download a ladder logic program to a MicroLogix controller, the software takes “file ownership” of the processor. File ownership prevents other devices from reading from or writing to the processor while the download is in process. If the controller were to respond to a device’s read commands during the download, the processor could respond with incorrect information. Similarly, if the controller were to accept information from other devices, the information could be lost because the program download sequence could immediately overwrite the information. Once the download is completed, the programming software returns the file ownership to the controller, so other devices can communicate with it again.

With the addition of DF1 half-duplex slave protocol, the controller clears the file ownership if no supported commands are received from the owner within the timeout period. If the file ownership were not cleared after a download sequence interruption, the processor would not accept commands from any other devices because it would assume another device still had file ownership.

If a download sequence is interrupted, due to noise caused by electromagnetic interference, discontinue communications to the controller for the *ownership timeout* period and restart the program download. The *ownership timeout* period is set to 60 seconds as a default for all protocols. However, if you are using DF1 half-duplex, and the *poll timeout* value is set to greater than 60 seconds, the *poll timeout* value will be used instead of the *ownership timeout*. After the timeout, you can re-establish communications with the processor and try the program download again. The only other way to clear file ownership is to cycle power on the processor.

Using Modems with MicroLogix 1000 Programmable Controllers

The types of modems that you can use with MicroLogix 1000 controllers include dial-up phone modems, leased-line modems, radio modems and line drivers. For point-to-point full-duplex modem connections that do not require any modem handshaking signals to operate, use DF1 full-duplex protocol. For point-to-multipoint modem connections, or for point-to-point modem connections that require Request-to-Send/Clear-To-Send (RTS/CTS) handshaking, use DF1 half-duplex slave protocol. In this case, one (and only one) of the other devices must be configured for DF1 half-duplex master protocol. Do not attempt to use DH-485 protocol through modems under any circumstance.

Note *Only Series D or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers support RTS/CTS modem handshaking and only when configured for DF1 half-duplex slave protocol with the control line parameter set to "Half-Duplex Modem". No other modem handshaking lines (i.e. Data Set Ready, Carrier Detect and Data Terminal Ready) are supported by any MicroLogix 1000 controllers.*

Dial-Up Phone Modems

Dial-up phone line modems support point-to-point full-duplex communications. Normally a MicroLogix 1000 controller, on the receiving end of the dial-up connection, will be configured for DF1 full-duplex protocol. The modem connected to the MicroLogix 1000 controller must support auto-answer and must not require any modem handshaking signals from the MicroLogix 1000 (i.e., DTR or RTS) in order to operate. The MicroLogix 1000 has no means to cause its modem to initiate or disconnect a phone call, so this must be done from the site of the remote modem.

Leased-Line Modems

Leased-line modems are used with dedicated phone lines that are typically leased from the local phone company. The dedicated lines may be in a point-to-point topology supporting full-duplex communications between two modems or in a point-to-multipoint topology supporting half-duplex communications between three or more modems. In the point-to-point topology, configure the MicroLogix 1000 controllers for DF1 full-duplex protocol (as long as the modems used do not require DTR or RTS to be high in order to operate). In the point-to-multipoint topology, configure the MicroLogix 1000 controllers for DF1 half-duplex slave protocol with the control line parameter set to "Half-Duplex Modem".

Radio Modems

Radio modems may be implemented in a point-to-point topology supporting either half-duplex or full-duplex communications, or in a point-to-multipoint topology supporting half-duplex communications between three or more modems. In the point-to-point topology using full-duplex radio modems, configure the MicroLogix 1000 controllers for DF1 full-duplex protocol (as long as the modems used do not require DTR or RTS to be high in order to operate). In the point-to-point topology using half-duplex radio modems, or point-to-multipoint topology using half-duplex radio modems, configure the MicroLogix 1000 controllers for DF1 half-duplex slave protocol. If these radio modems require RTS/CTS handshaking, configure the control line parameter to “Half-Duplex Modem”.

Line Drivers

Line drivers, also called short-haul “modems”, do not actually modulate the serial data, but rather condition the electrical signals to operate reliably over long transmission distances (up to several miles). Allen-Bradley’s AIC+ Advanced Interface Converter is a line driver that converts an RS-232 electrical signal into an RS-485 electrical signal, increasing the signal transmission distance from 50 to 4000 feet. In a point-to-point line driver topology, configure the MicroLogix 1000 controller for DF1 full-duplex protocol (as long as the line drivers do not require DTR or RTS to be high in order to operate). In a point-to-multipoint line driver topology, configure the MicroLgoix 1000 controllers for DF1 half-duplex slave protocol. If these line drivers require RTS/CTS handshaking, configure the control line parameter to “Half-Duplex Modem”.

DH-485 Communication Protocol

The information in this section describes the DH-485 network functions, network architecture, and performance characteristics. It will also help you plan and operate the MicroLogix 1000 on a DH-485 network.

Note *Only Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers support the DH-485 network.*

DH-485 Network Description

The DH-485 protocol defines the communication between multiple devices that co-exist on a single pair of wires. This protocol uses RS-485 half-duplex as its physical interface. (RS-485 is a definition of electrical characteristics; it is *not* a protocol.) RS-485 uses devices that are capable of co-existing on a common data circuit, thus allowing data to be easily shared between devices.

The DH-485 network offers:

- interconnection of 32 devices
- multi-master capability
- token passing access control
- the ability to add or remove nodes without disrupting the network
- maximum network length of 1219 m (4000 ft)

The DH-485 protocol supports two classes of devices: initiators and responders. All initiators on the network get a chance to initiate message transfers. To determine which initiator has the right to transmit, a token passing algorithm is used.

The following section describes the protocol used to control message transfers on the DH-485 network.

DH-485 Token Rotation

A node holding the token can send any valid packet onto the network. Each node is allowed only one transmission (plus two retries) each time it receives the token. After a node sends one message packet, it attempts to give the token to its successor by sending a “token pass” packet to its successor.

If no network activity occurs, the initiator sends the token pass packet again. After two retries (a total of three tries) the initiator will attempt to find a new successor.

The allowable range of the node address of an initiator is 0 to 31. The allowable address range for all responders is 1 to 31. There must be at least one initiator on the network.

DH-485 Configuration Parameters

When the system mode driver is DH-485 Master, the following parameters can be changed:

Parameter	Description	Default
Baud Rate	Toggles between the communication rate of 9600 and 19200.	19200
Node Address	This is the node address of the processor on the DH-485 network. The valid range is 1–31.	1
Max Node Address	This is the maximum node address of an active processor (fixed at 31). Set the node addresses of the devices on the network to low, sequential numbers for best performance.	31
Token Hold Factor	Determines the number of transactions allowed to make each DH-485 token rotation. (fixed at 1)	1

DH-485 Network Initialization

Network initialization begins when a period of inactivity exceeding the time of a link dead timeout is detected by an initiator on the network. When the time for a link dead timeout is exceeded, usually the initiator with the lowest address claims the token. When an initiator has the token it will begin to build the network. The network requires at least one initiator to initialize it.

Building a network begins when the initiator that claimed the token tries to pass the token to the successor node. If the attempt to pass the token fails, or if the initiator has no established successor (for example, when it powers up), it begins a linear search for a successor starting with the node above it in the addressing.

When the initiator finds another active initiator, it passes the token to that node, which repeats the process until the token is passed all the way around the network to the first node. At this point, the network is in a state of normal operation.

Devices that use the DH-485 Network

In addition to the Series C or later MicroLogix 1000 discrete controllers and all MicroLogix 1000 analog controllers, the devices shown in the following table also support the DH-485 network.

Note *You cannot connect the Hand-Held Programmer, 1761-HHP-B30, to the AIC+.*

Catalog Number	Description	Installation Requirement	Function	Publication
1747-L511, -L514, -L524, -L531, -L532 -L541, -L542, -L543 -L551, -L552 -L553	SLC 500 Processors	SLC Chassis	These processors support a variety of I/O requirements and functionality.	1747-6.2
1746-BAS	BASIC Module	SLC Chassis	Provides an interface for SLC 500 devices to foreign devices. Program in BASIC to interface the 3 channels (2 RS232 and 1 DH-485) to printers, modems, or the DH-485 network for data collection.	1746-6.1 1746-6.2 1746-6.3
1785-KA5	DH+™ /DH-485 Gateway	(1771) PLC Chassis	Provides communication between stations on the PLC-5® (DH+) and SLC 500 (DH-485) networks. Enables communication and data transfer from PLC® to SLC 500 on DH-485 network. Also enables programming software programming or data acquisition across DH+ to DH-485.	1785-6.5.5 1785-1.21
2760-RB	Flexible Interface Module	(1771) PLC Chassis	Provides an interface for SLC 500 (using protocol cartridge 2760-SFC3) to other A-B PLCs and devices. Three configurable channels are available to interface with Bar Code, Vision, RF, Dataliner™, and PLC systems.	2760-ND001
1784-KTX, -KTXD	PC DH-485 IM	IBM XT/AT Computer Bus	Provides DH-485 using RSLinx	1784-6.5.22
1784-PCMK	PCMCIA IM	PCMCIA slot in computer and Interchange	Provides DH-485 using RSLinx	1784-6.5.19

Catalog Number	Description	Installation Requirement	Function	Publication
1747-PT1	Hand-Held Terminal	NA	Provides hand-held programming, monitoring, configuring, and troubleshooting capabilities for SLC 500 processors.	1747-NP002
1747-DTAM, 2707-L8P1, -L8P2, -L40P1, -L40P2, -V40P1, -V40P2, -V40P2N, -M232P3, and -M485P3	DTAM, DTAM Plus, and DTAM Micro Operator Interfaces	Panel Mount	Provides electronic operator interface for SLC 500 processors.	1747-ND013 2707-800, 2707-803
2711-K5A2, -B5A2, -K5A5, -B5A5, -K5A1, -B5A1, -K9A2, -T9A2, -K9A5, -T9A5, -K9A1, and -T9A1	PanelView 550 and PanelView 900 Operator Terminals	Panel Mount	Provides electronic operator interface for SLC 500 processors.	2711-802, 2711-816

NA = Not Applicable

Important DH-485 Network Planning Considerations

Carefully plan your network configuration before installing any hardware. Listed below are some of the factors that can affect system performance:

- amount of electrical noise, temperature, and humidity in the network environment
- number of devices on the network
- connection and grounding quality in installation
- amount of communication traffic on the network
- type of process being controlled
- network configuration

The major hardware and software issues you need to resolve before installing a network are discussed in the following sections.

Hardware Considerations

You need to decide the length of the communication cable, where you route it, and how to protect it from the environment where it will be installed.

When the communication cable is installed, you need to know how many devices are to be connected during installation and how many devices will be added in the future. The following sections will help you understand and plan the network.

Number of Devices and Length of Communication Cable

You must install an AIC+ Advanced Interface Converter, catalog number 1761-NET-AIC, for each node on the network. If you plan to add nodes later, provide additional advanced interface converters during the initial installation to avoid recabling after the network is in operation.

The maximum length of the communication cable is 1219 m (4000 ft). This is the total cable distance from the first node to the last node on the network.

Planning Cable Routes

Follow these guidelines to help protect the communication cable from electrical interference:

- Keep the communication cable at least 1.52 m (5 ft) from any electric motors, transformers, rectifiers, generators, arc welders, induction furnaces, or sources of microwave radiation.
- If you must run the cable across power feed lines, run the cable at right angles to the lines.
- If you do not run the cable through a contiguous metallic wireway or conduit, keep the communication cable at least 0.15 m (6 in.) from ac power lines of less than 20A, 0.30 m (1 ft) from lines greater than 20A, but only up to 100k VA, and 0.60 m (2 ft) from lines of 100k VA or more.
- If you run the cable through a contiguous metallic wireway or conduit, keep the communication cable at least 0.08 m (3 in.) from ac power lines of less than 20A, 0.15 m (6 in.) from lines greater than 20A, but only up to 100k VA, and 0.30 m (1 ft) from lines of 100k VA or more.

  Running the communication cable through conduit provides extra protection from physical damage and electrical interference. If you route the cable through conduit, follow these additional recommendations:

  - Use ferromagnetic conduit near critical sources of electrical interference. You can use aluminum conduit in non-critical areas.
  - Use plastic connectors to couple between aluminum and ferromagnetic conduit. Make an electrical connection around the plastic connector (use pipe clamps and the heavy gauge wire or wire braid) to hold both sections at the same potential.
  - Ground the entire length of conduit by attaching it to the building earth ground.
  - Do not let the conduit touch the plug on the cable.
  - Arrange the cables loosely within the conduit. The conduit should contain only serial communication cables.
  - Install the conduit so that it meets all applicable codes and environmental specifications.

For more information on planning cable routes, see *Industrial Automation Wiring and Grounding Guidelines*, Publication Number 1770-4.1.

Software Considerations

Software considerations include the configuration of the network and the parameters that can be set to the specific requirements of the network. The following are major configuration factors that have a significant effect on network performance:

- number of nodes on the network
- addresses of those nodes
- baud rate

The following sections explain network considerations and describe ways to select parameters for optimum network performance (speed). See your programming software's user manual for more information.

Number of Nodes

The number of nodes on the network directly affects the data transfer time between nodes. Unnecessary nodes (such as a second programming terminal that is not being used) slow the data transfer rate. The maximum number of nodes on the network is 32.

Setting Node Addresses

The best network performance occurs when node addresses are assigned in sequential order. Initiators, such as personal computers, should be assigned the lowest numbered addresses to minimize the time required to initialize the network. The valid range for the MicroLogix 1000 controllers is 1–31 (controllers cannot be node 0). The default setting is 1. The node address is stored in the controller status file (S:16L).

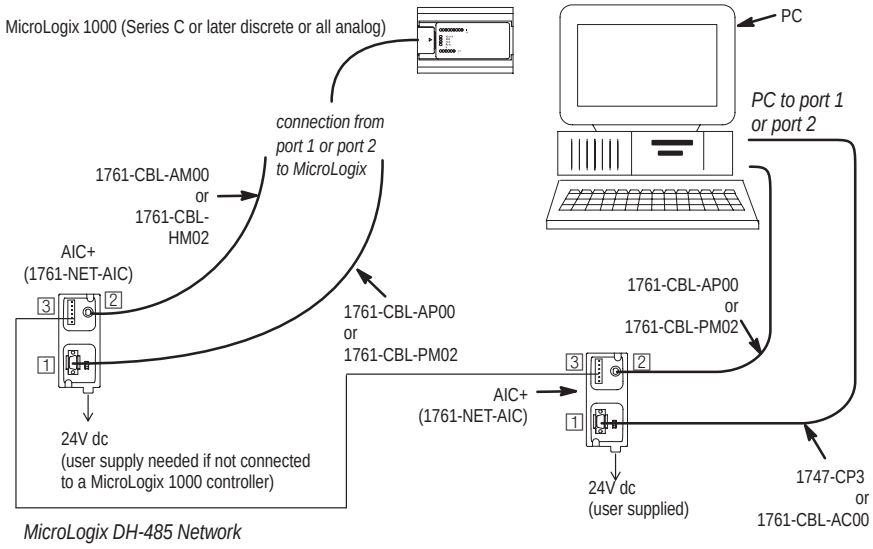
Setting Controller Baud Rate

The best network performance occurs at the highest baud rate, which is 19200. This is the default baud rate for a MicroLogix 1000 device on the DH-485 network. All devices must be at the same baud rate. This rate is stored in the controller status file (S:16H).

Example DH-485 Connections

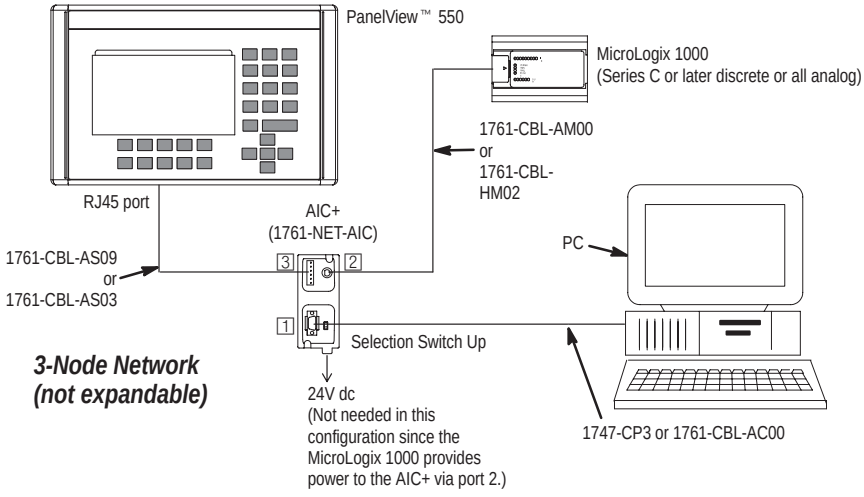
The following network diagrams provide examples of how to connect Series C or later MicroLogix 1000 discrete and all MicroLogix 1000 analog controllers to the DH-485 network using the AIC+. For more information on the AIC+, see the *Advanced Interface Converter and DeviceNet Interface Installation Instructions*, Publication 1761-5.11.

DH-485 Network with a MicroLogix 1000 Controller



MicroLogix DH-485 Network

Typical 3-Node Network

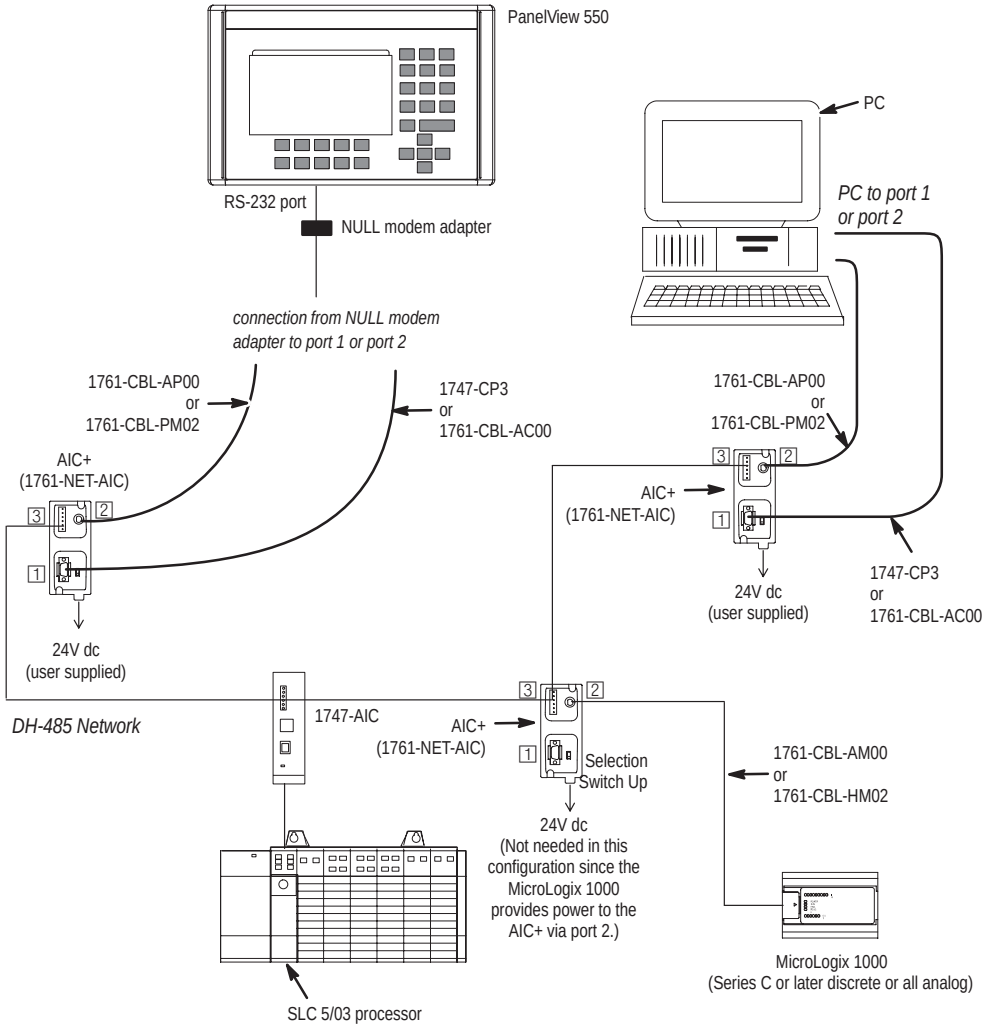


1 DB-9 RS-232 port

2 mini-DIN 8 RS-232 port

3 DH-485/DF1 port

Networked Operator Interface Device and MicroLogix Controller


PanelView 550
RS-232 port
NULL modem adapter
connection from NULL modem adapter to port 1 or port 2
PC
PC to port 1 or port 2
1761-CBL-AP00 or 1761-CBL-PM02
1747-CP3 or 1761-CBL-AC00
AIC+ (1761-NET-AIC)
24V dc (user supplied)
1761-CBL-AP00 or 1761-CBL-PM02
AIC+ (1761-NET-AIC)
24V dc (user supplied)
1747-CP3 or 1761-CBL-AC00
DH-485 Network
1747-AIC
AIC+ (1761-NET-AIC)
Selection Switch Up
24V dc (Not needed in this configuration since the MicroLogix 1000 provides power to the AIC+ via port 2.)
1761-CBL-AM00 or 1761-CBL-HM02
MicroLogix 1000 (Series C or later discrete or all analog)
SLC 5/03 processor

[1] DB-9 RS-232 port

[2] mini-DIN 8 RS-232 port

[3] DH-485/DF1 port

Reference

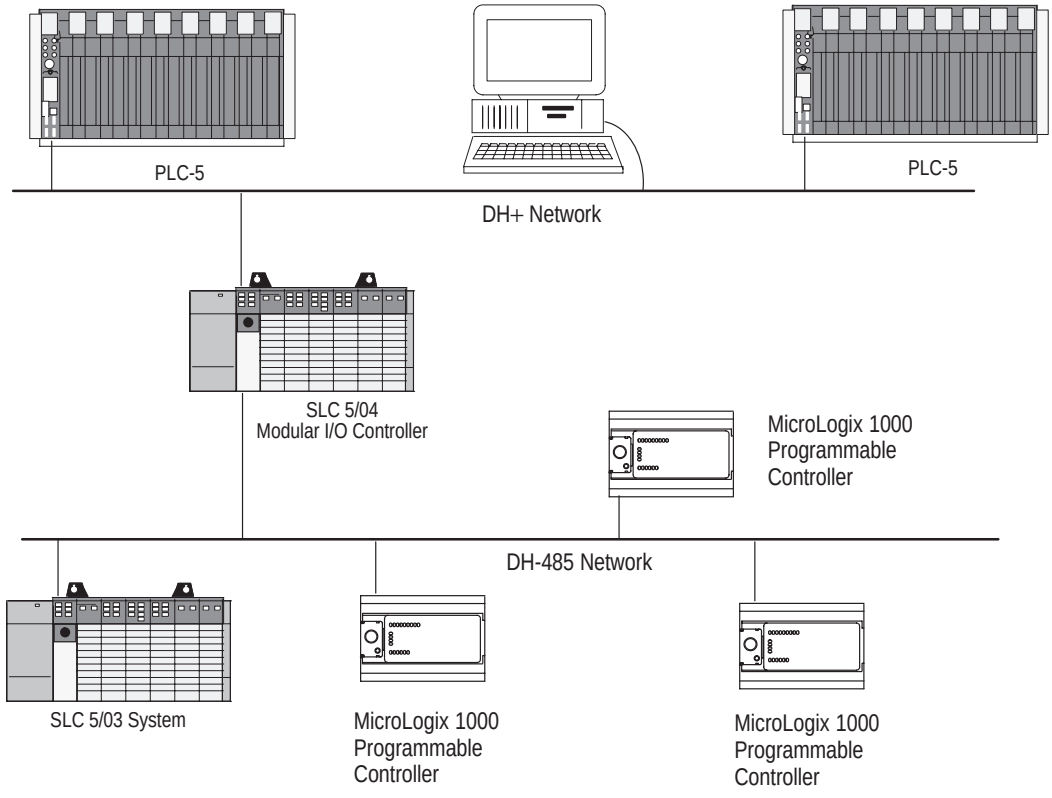
MicroLogix Remote Packet Support

Series D MicroLogix controllers and all MicroLogix analog controllers can respond to communication packets (or commands) that do not originate on the local DH-485 network. This is useful in installations where communication is needed between the DH-485 and DH+ networks.

The example below shows how to send messages from a PLC device or a PC on the DH+ network to a MicroLogix 1000 controller on the DH-485 network. This method uses an SLC 5/04 processor bridge connection.

When using this method:

- PLC-5 devices can send read and write commands to MicroLogix controllers.
- MicroLogix 1000 controllers can respond to MSG instructions received. The MicroLogix controllers cannot initiate MSG instructions to devices on the DH+ network.
- PC can send read and write commands to MicroLogix controllers.
- PC can do remote programming of MicroLogix controllers.



E Application Example Programs

This appendix is designed to illustrate various instructions described previously in this manual. Application example programs include:

- paper drilling machine using most of the software instructions
- time driven sequencer using TON and SQO instructions
- event driven sequencer using SQC and SQO instructions
- bottle line example using the HSC instruction (Up/down counter)
- pick and place machine example using the HSC instruction (Quadrature Encoder with reset and hold)
- RPM calculation using HSC, RTO, timer, and math instructions
- on/off circuit using basic, program flow, and application specific instructions
- spray booth using bit shift and FIFO instructions
- adjustable time delay example using timer instructions

Because of the variety of uses for this information, the user of and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the program. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use of application of this information.

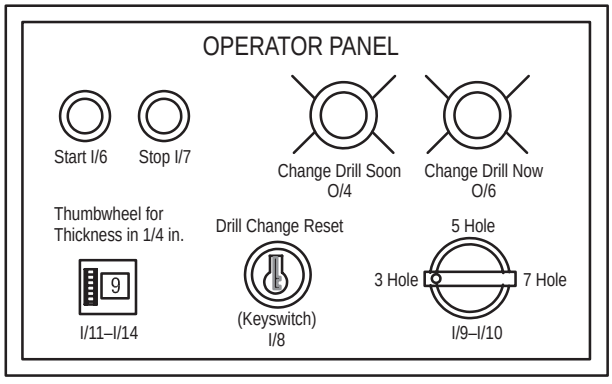
The illustrations, charts, and examples shown in this appendix are intended solely to illustrate the principles of the controller and some of the methods used to apply them. Particularly because of the many requirements associated with any particular installation, Allen-Bradley Company cannot assume responsibility or liability for actual use based upon the illustrative uses and applications.

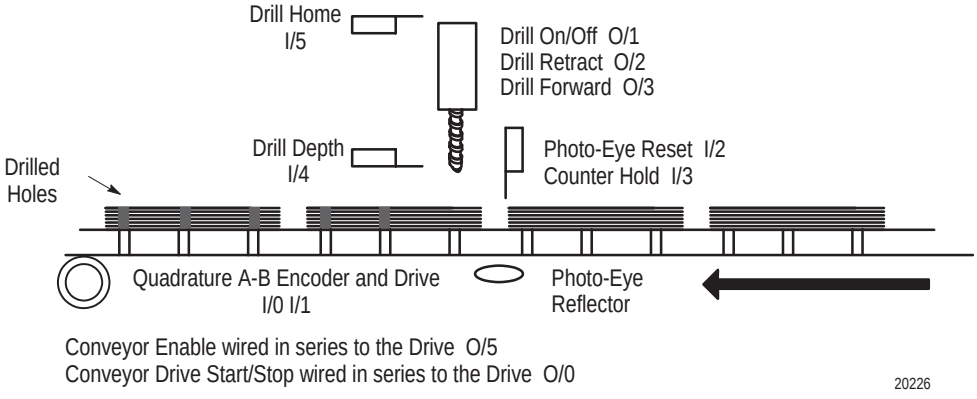
Paper Drilling Machine Application Example

For a detailed explanation of:

- XIC, XIO, OTE, RES, OTU, OTL, and OSR instructions, see chapter 6.
- EQU and GEQ instructions, see chapter 7.
- CLR, ADD, and SUB instructions, see chapter 8.
- MOV and FRD instructions, see chapter 9.
- JSR and RET instructions, see chapter 10.
- INT and SQO instructions, see chapter 11.
- HSC, HSL, and RAC instructions, see chapter 12.

This machine can drill 3 different hole patterns into bound manuals. The program tracks drill wear and signals the operator that the bit needs replacement. The machine shuts down if the signal is ignored by the operator.





Paper Drilling Machine Operation Overview

Undrilled books are placed onto a conveyor taking them to a single spindle drill. Each book moves down the conveyor until it reaches the first drilling position. The conveyor stops moving and the drill lowers and drills the first hole. The drill then retracts and the conveyor moves the same book to the second drilling position. The drilling process is repeated until there are the desired holes per book.

Drill Mechanism Operation

When the operator presses the start button, the drill motor turns on. After the book is in the first drilling position, the conveyor subroutine sets a drill sequence start bit, and the drill moves toward the book. When the drill has drilled through the book, the drill body hits a limit switch and causes the drill to retract up out of the book. When the drill body is fully retracted, the drill body hits another limit switch indicating that it is in the home position. Hitting the second limit switch unlatches the drill sequence start bit and causes the conveyor to move the book to the next drilling position.

Conveyor Operation

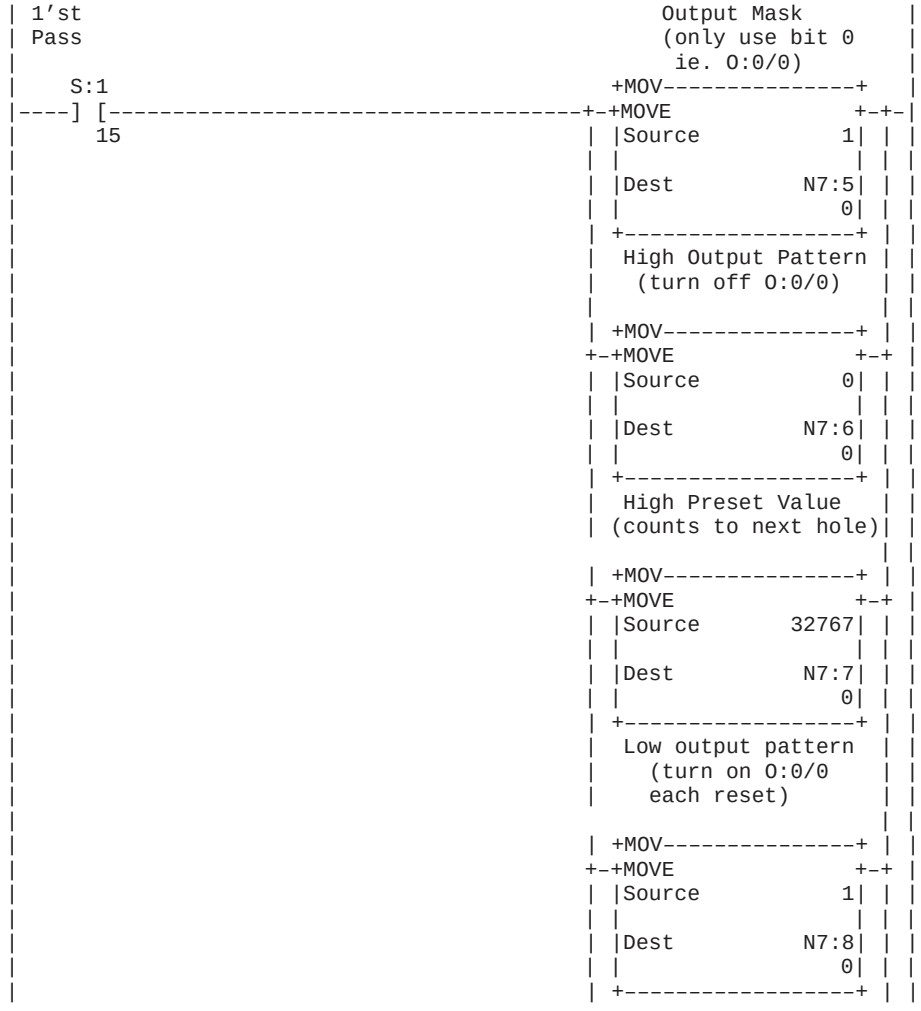
When the start button is pressed, the conveyor moves the books forward. As the first book moves close to the drill, the book trips a photo-eye sensor. This tells the machine where the leading edge of the book is. Based on the position of the selector switch, the conveyor moves the book until it reaches the first drilling position. The drill sequence start bit is set and the first hole is drilled. The drill sequence start bit is now unlatched and the conveyor moves the same book to the second drilling position. The drilling process is repeated until there are the desired holes per book. The machine then looks for another book to break the photo-eye beam and the process is repeated. The operator can change the number of drilled holes by changing the selector switch.

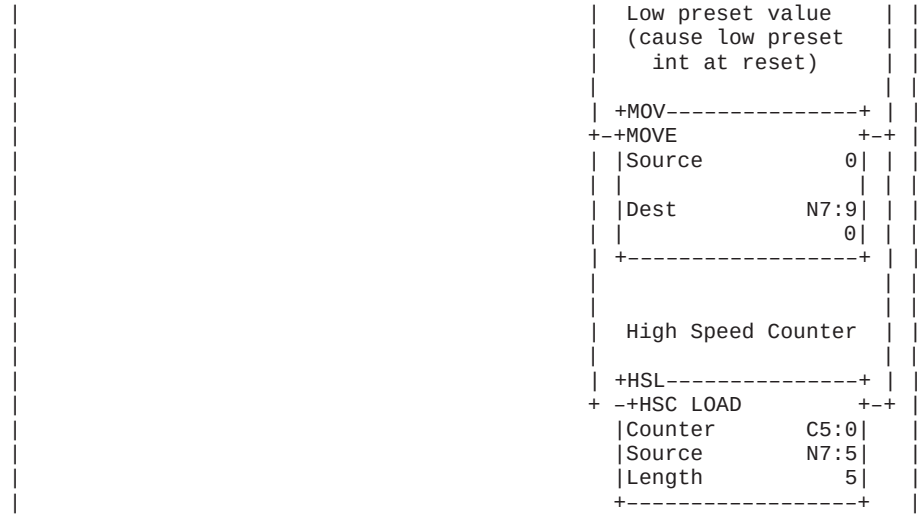
Drill Calculation and Warning

The program tracks the number of holes drilled and the number of inches of material that have been drilled through using a thumbwheel. The thumbwheel is set to the thickness of the book per 1/4 inch. (If the book is 1 1/2 inches thick, the operator would set the thumbwheel to 6.) When 25,000 inches have been drilled, the Change Drill Soon pilot light turns on. When 25,500 inches have been drilled, the Change Drill Soon pilot light flashes. When 26,000 inches have been drilled, the Change Drill Now pilot light turns on and the machine turns off. The operator changes drill bits and then resets the internal drill wear counter by turning the Drill Change Reset keyswitch.

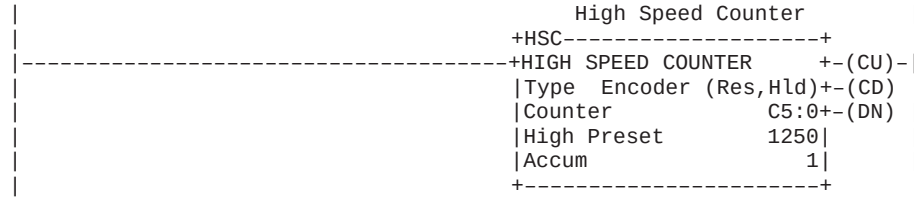
Paper Drilling Machine Ladder Program

Rung 2:0
Initializes the high-speed counter each time the REM Run mode is entered. The high-speed counter data area (N7:5 – N7:9) corresponds with the starting address (source address) of the HSL instruction. The HSC instruction is disabled each entry into the REM run mode until the first time that it is executed as true. (The high preset was ”pegged” on initialization to prevent a high preset interrupt from occurring during the initialization process.)

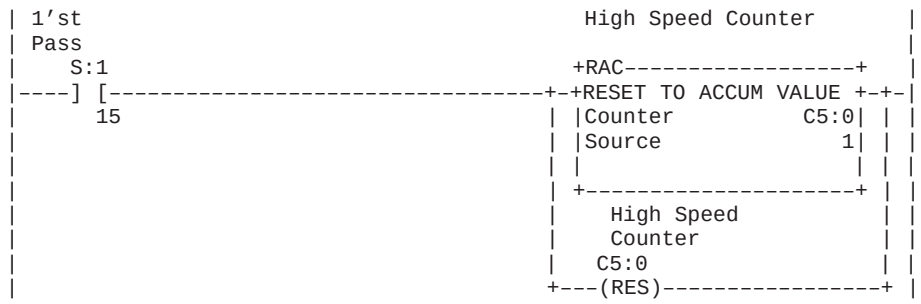




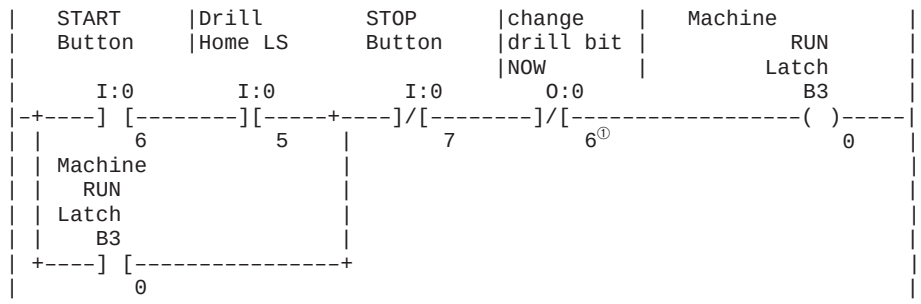
Rung 2:1
This HSC instruction is not placed in the high-speed counter interrupt subroutine. If this instruction were placed in the interrupt subroutine, the high-speed counter could never be started or initialized (because an interrupt must first occur in order to scan the high-speed counter interrupt subroutine).



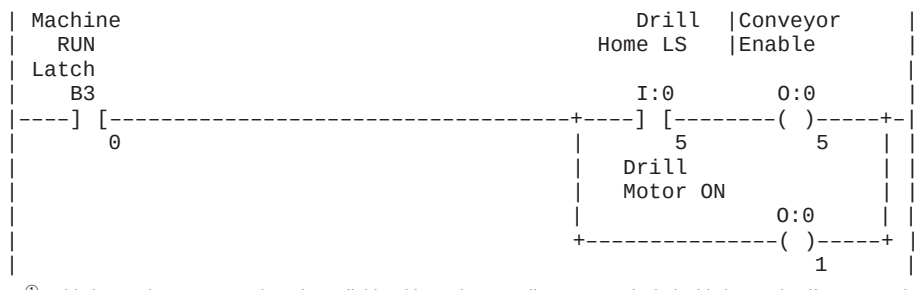
Rung 2:2
Forces a high-speed counter low preset interrupt to occur each REM Run mode entry. An interrupt can only occur on the transition of the high-speed counter accum to a preset value (accum reset to 1, then 0). This is done to allow the high-speed counter interrupt subroutine sequencers to initialize. The order of high-speed counter initialization is: (1)load high-speed counter parameters (2)execute HSL instruction (3)execute true HSC instruction (4)(optional) force high-speed counter interrupt to occur.



Rung 2:3
Starts the conveyor in motion when the start button is pressed. However, another condition must also be met before we start the conveyor: the drill bit must be in its fully retracted position (home). This rung also stops the conveyor when the stop button is pressed.

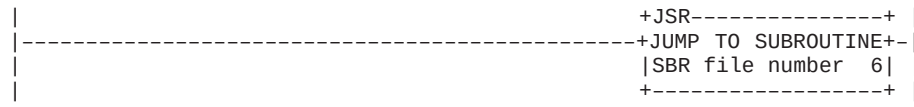


Rung 2:4
Applies the above start logic to the conveyor and drill motor.

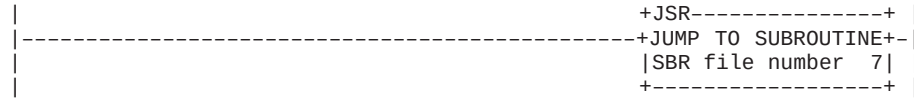


① This instruction accesses I/O only available with 32 I/O controllers. Do not include this instruction if you are using a 16 I/O controller.

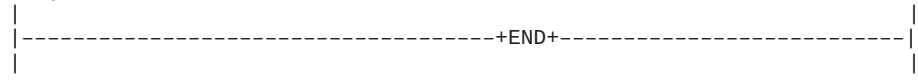
Rung 2:5
Calls the drill sequence subroutine. This subroutine manages the operation of a drilling sequence and restarts the conveyor upon completion of the drilling sequence



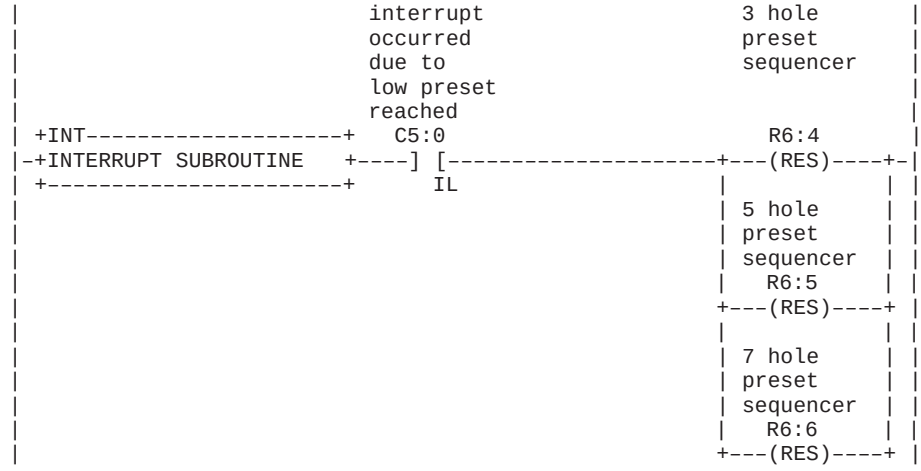
Rung 2:6
Calls the subroutine that tracks the amount of wear on the current drill bit.



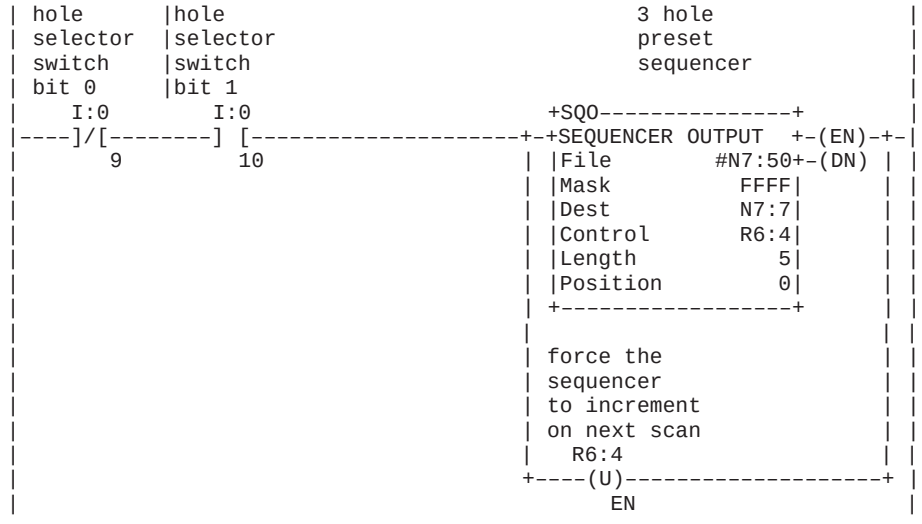
Rung 2:7



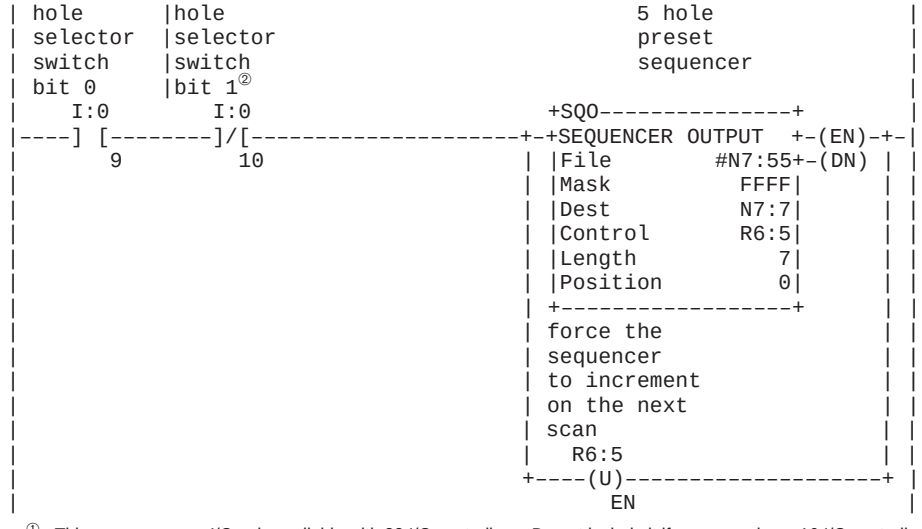
Rung 4:0
Resets the hole count sequencers each time that the low preset is reached. The low preset has been set to zero to cause an interrupt to occur each time that a reset occurs. The low preset is reached anytime that a reset C5:0 or hardware reset occurs. This ensures that the first preset value is loaded into the high-speed counter at each entry into the REM Run mode and each time that the external reset signal is activated.



Rung 4:1[1]
Keeps track of the hole number that is being drilled and loads the correct high-speed counter preset based on the hole count. This rung is only active when the "hole selector switch" is in the "3-hole" position. The sequencer uses step 0 as a null step upon reset. It uses the last step as a "go forever" in anticipation of the "end of manual" hard wired external reset.



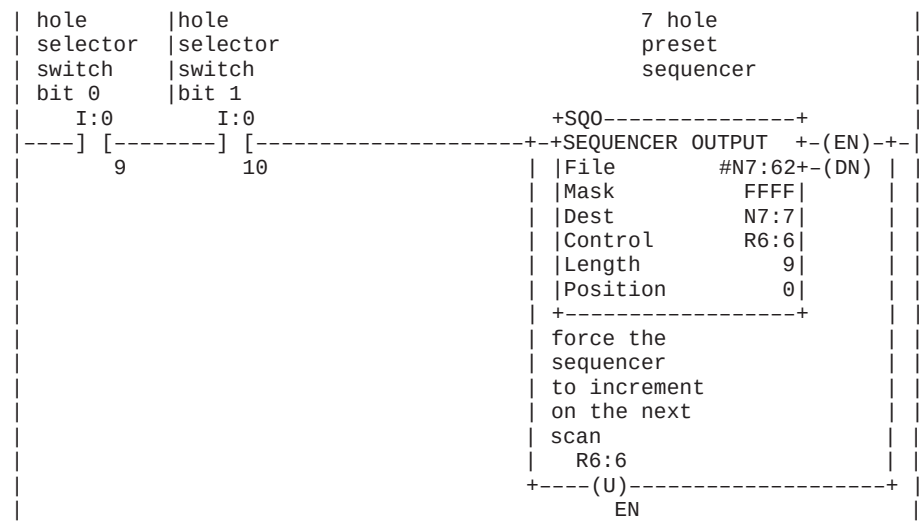
Rung 4:2
Is identical to the previous rung except that it is only active when the "hole selector switch" is in the "5-hole" position.



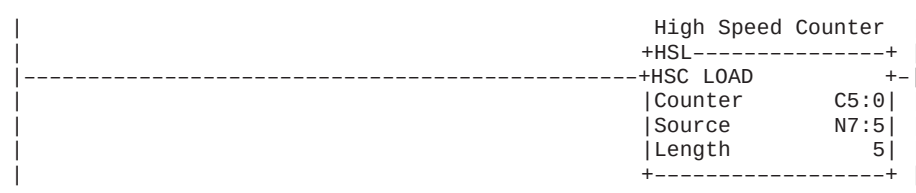
[1] This rung accesses I/O only available with 32 I/O controllers. Do not include it if you are using a 16 I/O controller.

[2] This instruction accesses I/O only available with 32 I/O controllers. Do not include it if you are using a 16 I/O controller.

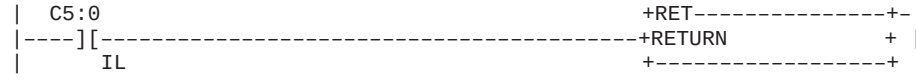
Rung 4:3[1]
Is identical to the 2 previous rungs except that it is only active when the ”hole selector switch” is in the ”7-hole” position.



Rung 4:4
Ensures that the high-speed counter preset value (N7:7) is immediately applied to the HSC instruction.



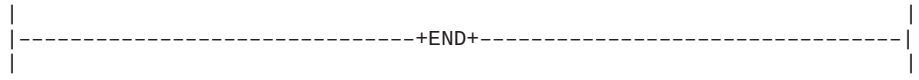
Rung 4:5
Interrupt occurred due to low preset reached.



[1] This rung accesses I/O only available with 32 I/O controllers. Do not include this rung if you are using a 16 I/O controller.

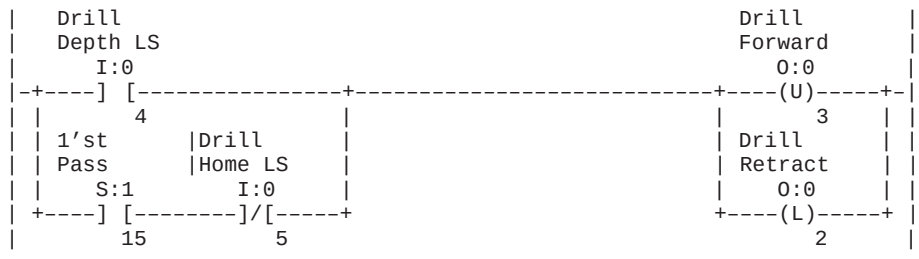
Rung 4:6
Signals the main program (file 2) to initiate a drilling sequence. The high-speed counter has already stopped the conveyor at the correct position using its high preset output pattern data (clear O:0/0). This occurred within microseconds of the high preset being reached (just prior to entering this high-speed counter interrupt subroutine). The drill sequence subroutine resets the drill sequence start bit and sets the conveyor drive bit (O:0/0) upon completion of the drilling sequence.

Rung 4:7

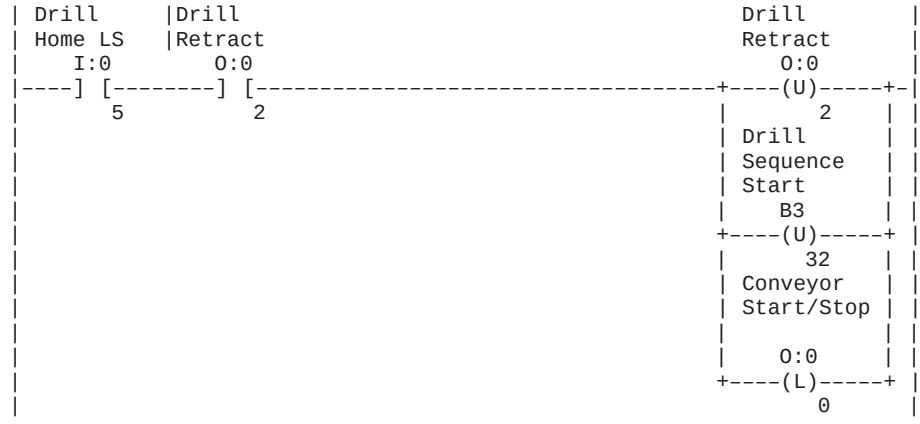


Rung 6:0
This section of ladder logic controls the up/down motion of the drill for the book drilling machine. When the conveyor positions the book under the drill, the DRILL SEQUENCE START bit is set. This rung uses that bit to begin the drilling operation. Because the bit is set for the entire drilling operation, the OSR is required to be able to turn off the forward signal so the drill can retract.

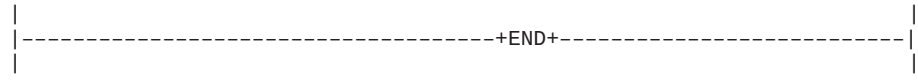
Rung 6:1
When the drill has drilled through the book, the body of the drill actuates the DRILL DEPTH limit switch. When this happens, the DRILL FORWARD signal is turned off and the DRILL RETRACT signal is turned on. The drill is also retracted automatically on power up if it is not actuating the DRILL HOME limit switch.



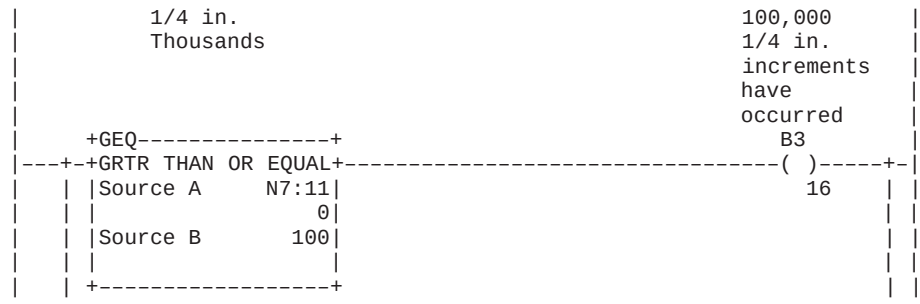
Rung 6:2
When the drill is retracting (after drilling a hole), the body of the drill actuates the DRILL HOME limit switch. When this happens the DRILL RETRACT signal is turned off, the DRILL SEQUENCE START bit is turned off to indicate the drilling process is complete, and the conveyor is restarted.

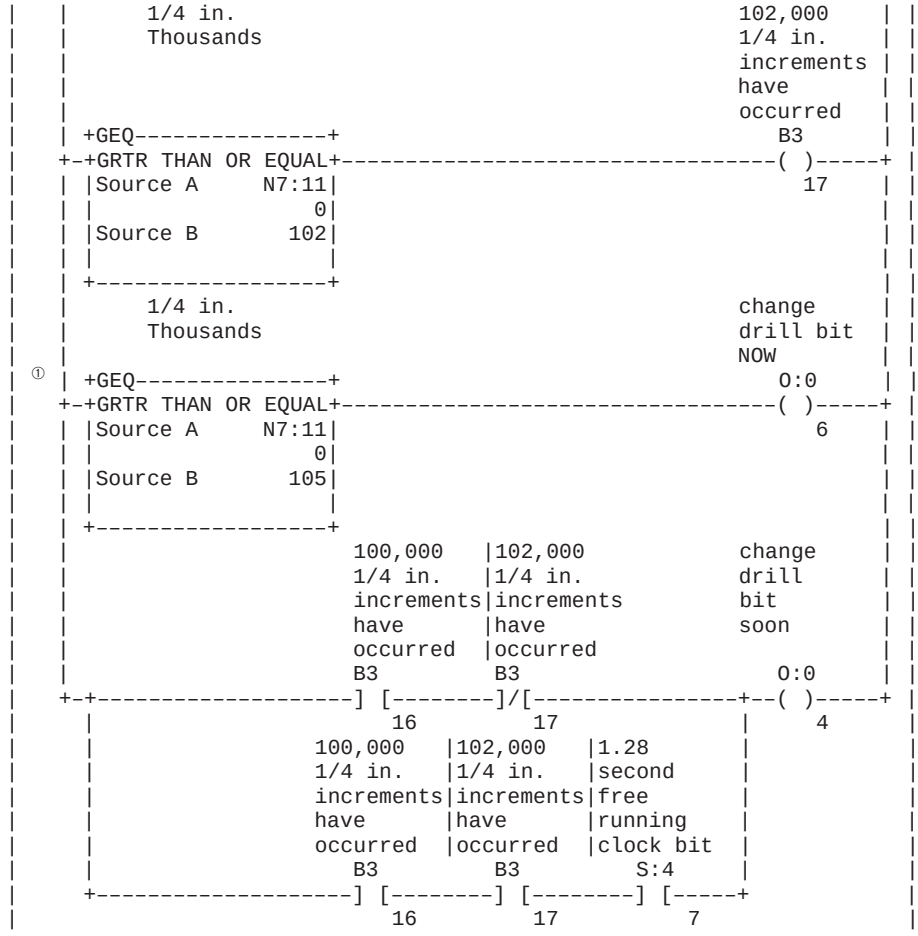


Rung 6.3



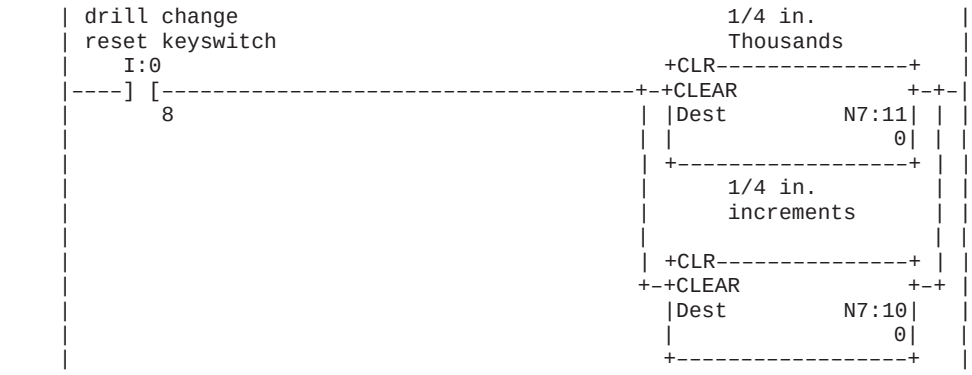
Rung 7:0
Examines the number of 1/4 in. thousands that have accumulated over the life of the current drill bit. If the bit has drilled between 100,000–101,999 1/4 in. increments of paper, the ”change drill” light illuminates steadily. When the value is between 102,000–103,999, the ”change drill” light flashes at a 1.28 second rate. When the value reaches 105,000, the ”change drill” light flashes, and the ”change drill now” light illuminates.



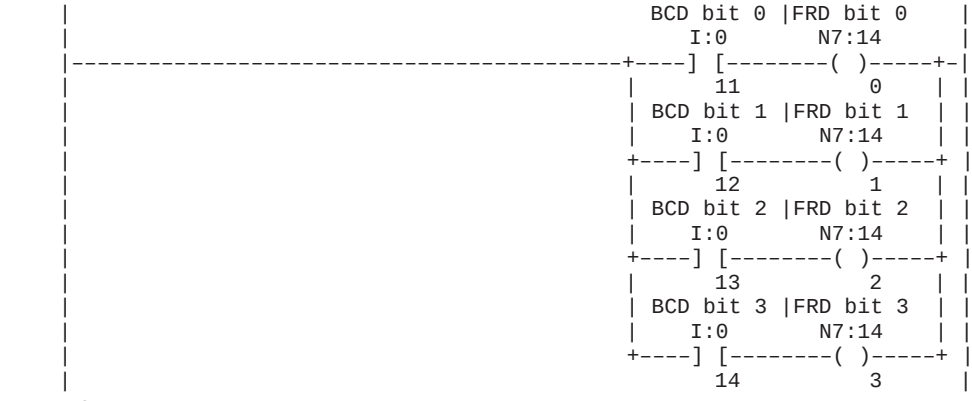


① This branch accesses I/O only available with 32 I/O controllers. Do not include this branch if you are using a 16 I/O controller.

Rung 7:1
Resets the number of 1/4 in. increments and the 1/4 in. thousands when the ”drill change reset” keyswitch is energized. This should occur following each drill bit change.

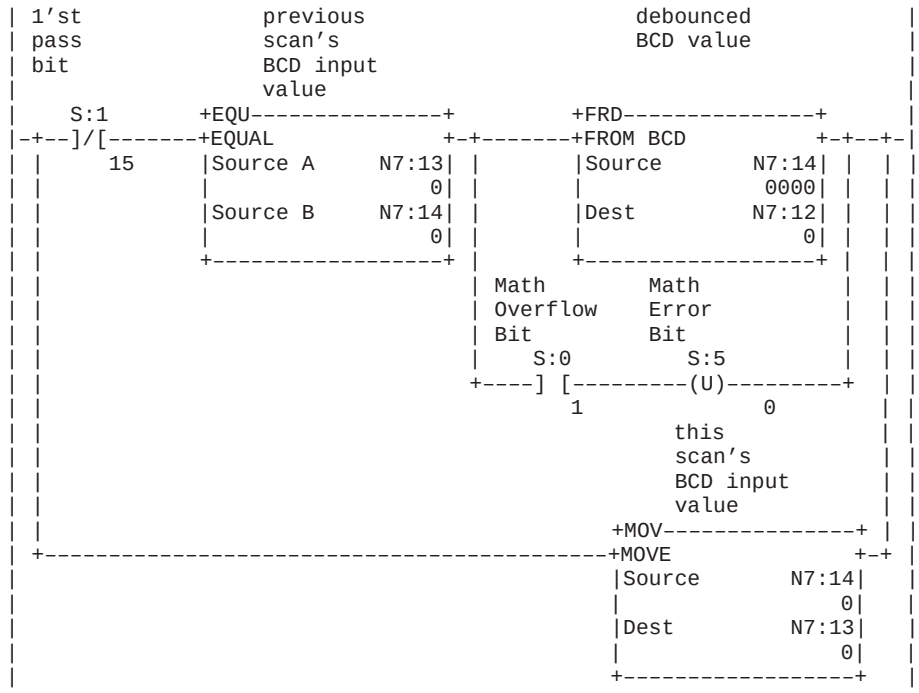


Rung 7:2[1]
Moves the single digit BCD thumbwheel value into an internal integer register. This is done to properly align the four BCD input signals prior to executing the BCD to Integer instruction (FRD). The thumbwheel is used to allow the operator to enter the thickness of the paper that is to be drilled. The thickness is entered in 1/4 in. increments. This provides a range of 1/4 in. to 2.25 in.

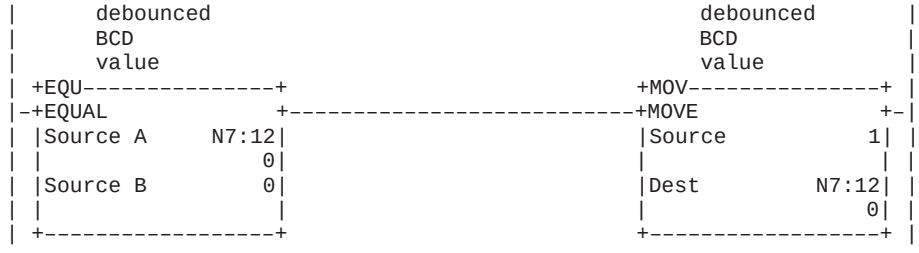


[1] This rung accesses I/O only available with 32 I/O controllers. Do not include this rung if you are using a 16 I/O controller.

Rung 7:3
Converts the BCD thumbwheel value from BCD to integer. This is done because the controller operates upon integer values. This rung also "debounces" the thumbwheel to ensure that the conversion only occurs on valid BCD values. Note that invalid BCD values can occur while the operator is changing the BCD thumbwheel. This is due to input filter propagation delay differences between the 4 input circuits that provide the BCD input value.

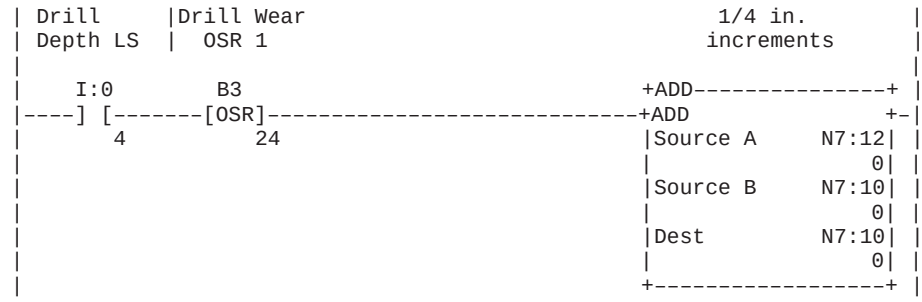


Rung 7:4
Ensures that the operator cannot select a paper thickness of 0. If this were allowed, the drill bit life calculation could be defeated resulting in poor quality holes due to a dull drill bit. Therefore the minimum paper thickness used to calculate drill bit wear is 1/4 in.

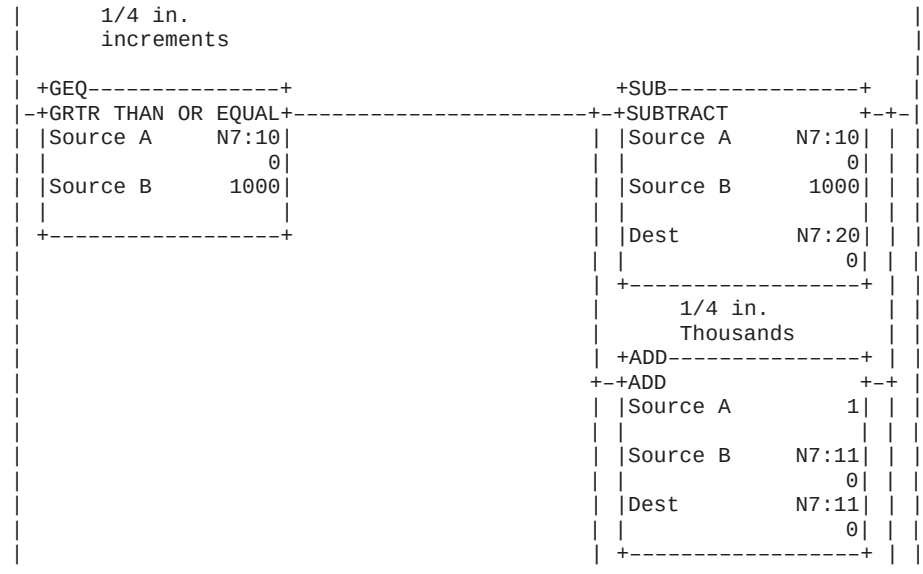


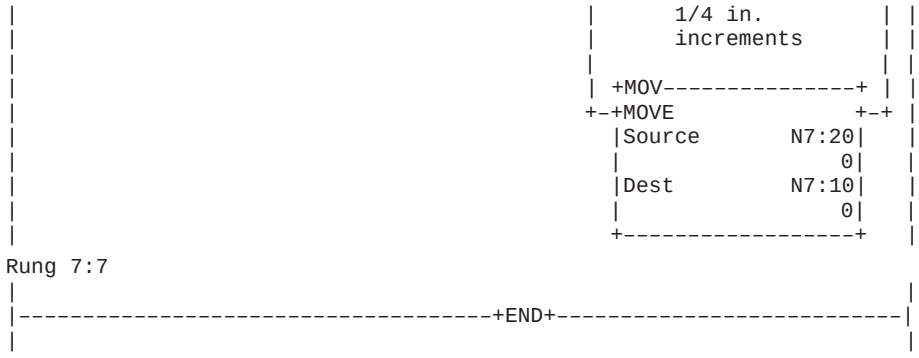
Reference

Rung 7:5
Keeps a running total of how many inches of paper have been drilled with the current drill bit. Every time a hole is drilled, adds the thickness (in 1/4 ins) to the running total (kept in 1/4 ins). The OSR is necessary because the ADD executes every time the rung is true, and the drill body would actuate the DRILL DEPTH limit switch for more than 1 program scan. Integer N7:12 is the integer-converted value of the BCD thumbwheel on inputs I:0/11 – I:0/14.



Rung 7:6
When the number of 1/4 in. increments surpasses 1000, finds out now many increments are past 1000 and stores in N7:20. Add 1 to the total of ’1000 1/4 in.’ increments, and re-initializes the 1/4 in. increments accumulator to how many increments were beyond 1000.





Time Driven Sequencer Application Example

The following application example illustrates the use of the TON and SQO instructions in a traffic signal at an intersection. The timing requirements are:

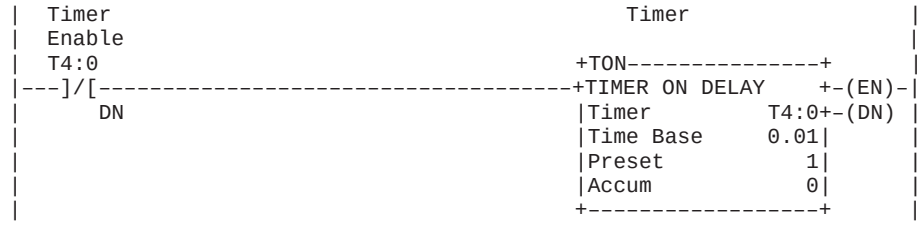
- Red light – 30 seconds
- Yellow light – 15 seconds
- Green light – 60 seconds

The timer, when it reaches its preset, steps the sequencer that in turn controls which traffic signal is illuminated. For a detailed explanation of:

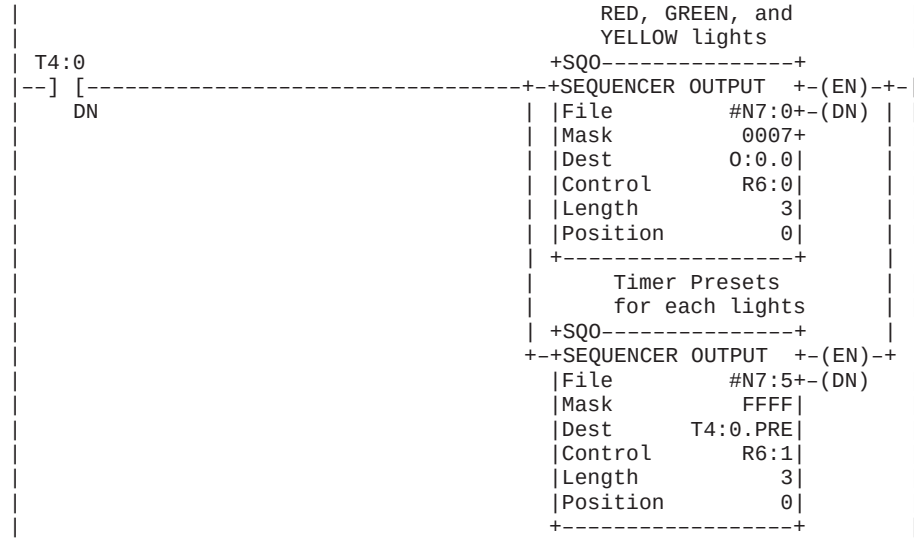
- XIC, XIO, and TON instructions, see chapter 6.
- SQO and SQC instructions, see chapter 11.

Time Driven Sequencer Ladder Program

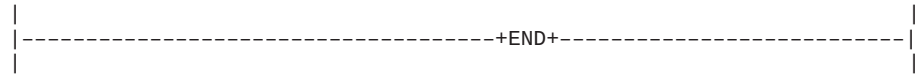
Rung 2:0
The function of this rung is called a regenerative timer. Every time the timer reaches its preset, the DONE bit is set for one scan – this causes this rung to become FALSE for one scan and resets the timer. On the following scan, when this rung becomes TRUE again, the timer begins timing.



Rung 2:1
Controls the RED, GREEN, and YELLOW lights wired to outputs O:0/0 – O:0/2, and controls how long the regenerative timer times between each step. When this rung goes from false-to-true (by the timer reaching its preset), the first sequencer changes which traffic light is illuminated, and the second sequencer changes the preset of the timer to determine how long this next light is illuminated.



Rung 2.2



```
Data Files

Address    15          Data           0
N7:0       0000    0000    0000    0000
N7:1       0000    0000    0000    0100
N7:2       0000    0000    0000    0010
N7:3       0000    0000    0000    0001


Data Table

Address    Data    (Radix=Decimal)

N7:0       0   4   2   1   0   0   6000    1500    3000
```

Event Driven Sequencer Application Example

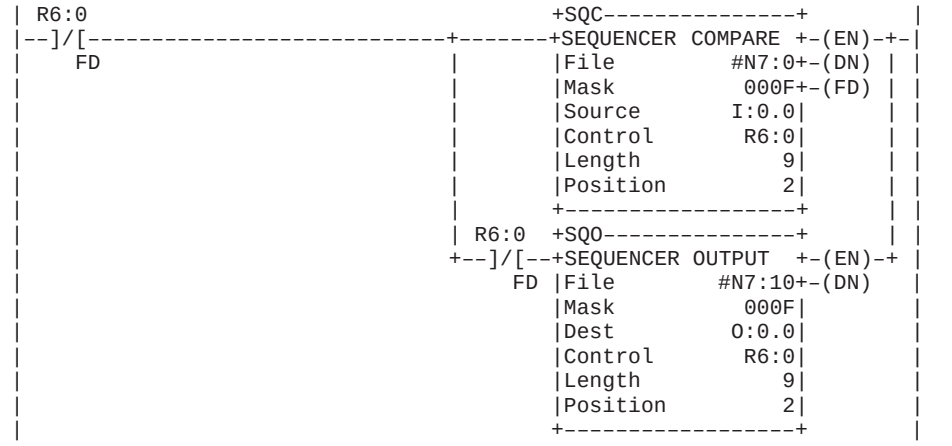
The following application example illustrates how the FD (found) bit on an SQC instruction can be used to advance an SQO to the next step (position). This application program is used when a specific order of events is required to occur repeatedly. By using this combination, you can eliminate using the XIO, XIC, and other instructions. For a detailed explanation of:

- XIC, XIO, and RES instructions, see chapter 6.
- SQO and SQC instructions, see chapter 11.

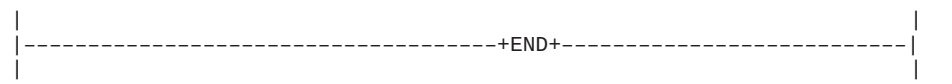
Event Driven Sequencer Ladder Program

```
Rung 2:0
Ensures that the SQO always resets to step (position) 1 each REM Run
mode entry.  (This rung actually resets the control register's position
and EN enable bit to 0. Due to this the following rung sees a false to
true transition and asserts step (position) 1 on the first scan.)

Eliminate this rung for retentive operation.
|  S:1                                                    R6:0     |
|--] [-------------------------------------------------------(RES)----|
|    15                                                            |
|                                                                  |
```

Rung 2:1
The SQC instruction and SQO instruction share the same Control Register. This is acceptable due to the careful planning of the rungstate condition. You could cascade (branch) many more SQO instructions below the SQO if you desired, all using the same Control Register (R6:0 in this case). Notice that we are only comparing Inputs 0–3 and are only asserting Outputs 0–3 (per our Mask value).



Rung 2.2



The following displays the FILE DATA for both sequencers. The SQC compare data starts at N7:0 and ends at N7:9. While the SQO output data starts at N7:10 and ends at N7:19. Please note that step 0 of the SQO is never active. The reset rung combined with the rung logic of sequencers guarantees that the sequencers always start at step 1. Both sequencers also ″roll over″ to step 1. ″Roll Over″ to step 1 is integral to all sequencer instructions.

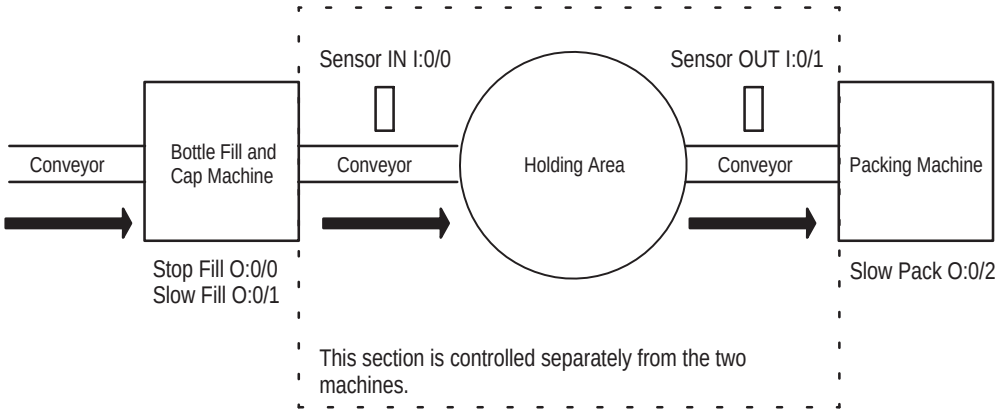
SQC Compare Data

Addresses	Data								
	(Radix=Decimal)								
N7:0	0	1	2	3	4	5	6	7	8	9
N7:10	0	0	1	2	3	4	5	6	7	8

Bottle Line Example

The following application example illustrates how the controller high-speed counter is configured for an Up/down counter. For a detailed explanation of:

- XIC, OTL, OTU and OTE instructions, see chapter 6.
- GRT, LES, and GEQ instruction, see chapter 7.
- HSC and HSL instructions, see chapter 12.



Bottle Line Operation Overview

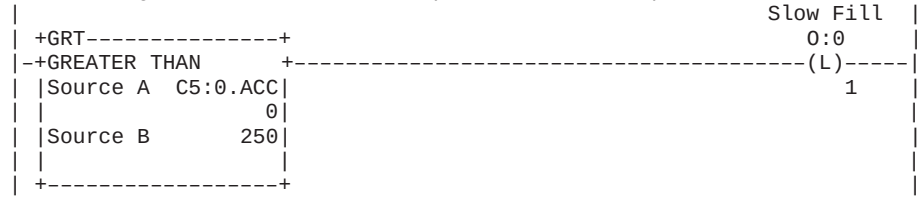
The controller on the conveyor, within the specified area above, regulates the speeds of the bottle fill and packing machines. Each machine is connected to a separate controller that communicates with the conveyor controller. The following ladder program is for the conveyor controller.

A conveyor feeds filled bottles past a proximity sensor (IN) to a holding area. The proximity sensor is wired to the I/0 terminal (up count) of the conveyor controller. The bottles are then sent on another conveyor past a proximity switch (OUT) to the packing machine. This proximity switch is wired to the I/1 terminal (down count) on the same controller.

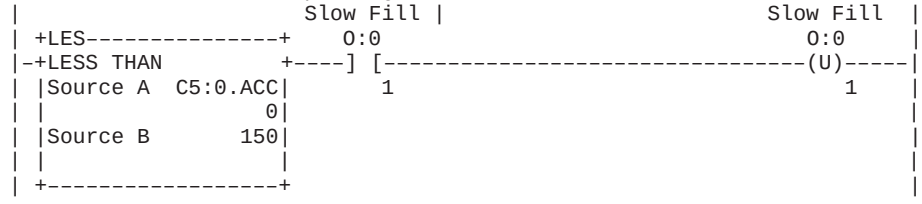
Bottle Line Ladder Program

```
Rung 2:0
Loads the high-speed counter with the following parameters:
N7:0 – 0001h Output Mask – Effect only O:0/0
N7:1 – 0001h Output Pattern for High Preset – Energize O:0/0 upon high
preset
N7:2 – 350d High Preset – Maximum numbers of bottles for the holding
area
N7:3 – 0000h Output Pattern for Low Preset – not used
N7:4 – 0d Low Preset – not used
| First Pass                                                          |
|    Bit                                                              |
|    S:1                                        +HSL----------------+ |
|----] [----------------------------------------+HSC LOAD          +-|
|      15                                       |Counter       C5:0| |
|                                               |Source        N7:0| |
|                                               |Length           5| |
|                                               +------------------+ |
Rung 2:1
Starts up the high-speed counter with the above parameters.  Each time
the rung is evaluated, the hardware accumulator is written to C5:0.ACC.
|                                             +HSC----------------+      |
|---------------------------------------------+HIGH SPEED COUNTER+-(CU)-|
|                                             |Type       Up/Down+-(CD) |
|                                             |Counter       C5:0+-(DN) |
|                                             |Preset         350|      |
|                                             |Accum            0|      |
|                                             +------------------+      |
Rung 2:2
Packing machine running too fast for the filling machine.  Slow down
the packing machine to allow the filler to catch up.
|                                                          Slow Pack  |
| +LES----------------+                                       O:0     |
|-+LESS THAN          +-----------------------------------------(L)-----|
| |Source A  C5:0.ACC|                                          2     |
| |                 0|                                                |
| |Source B       100|                                                |
| |                  |                                                |
| +------------------+                                                |
Rung 2:3
If the packer was slowed down to allow the filler to catch up, wait
until the holding area is approximately 2/3 full before allowing the
packer to run at full speed again.
|                     Slow Pack |                          Slow Pack  |
| +GRT----------------+    O:0                               O:0     |
|-+GREATER THAN       +----] [---------------------------------(U)-----|
| |Source A  C5:0.ACC|       2                                  2    |
| |                 0|                                               |
| |Source B       200|                                               |
| |                  |                                               |
| +------------------+                                               |
```

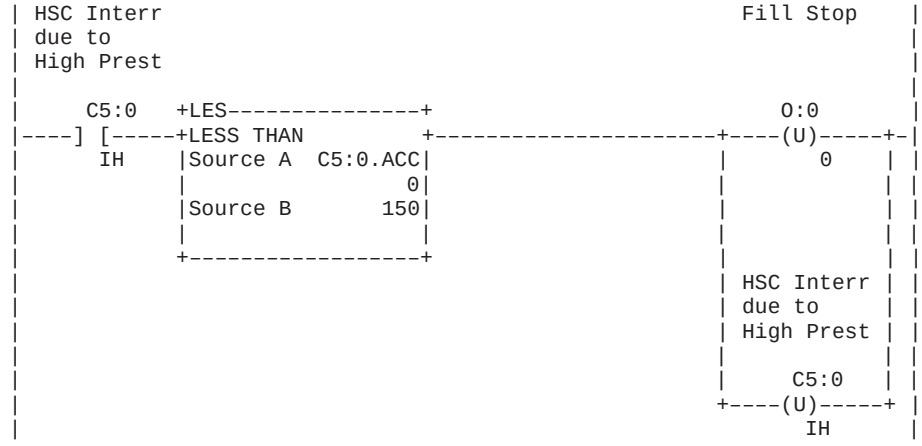
Rung 2:4
Filling machine running too fast for the packing machine. Slow down the filling machine to allow the packer to catch up.



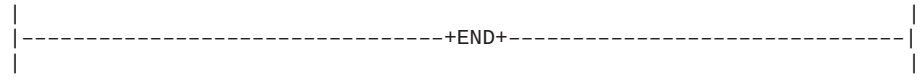
Rung 2:5
If the filler was slowed down to allow the packer to catch up, wait until the holding area is approximately 1/3 full before allowing the filler to run at full speed again.



Rung 2:6
If the high-speed counter reached its high preset of 350 (indicates that the holding area reached maximum capacity), it would energize O:0/0, shutting down the filling operation. Before re-starting the filler, allow the packer to empty the holding area until it is about 1/3 full.



Rung 2:7

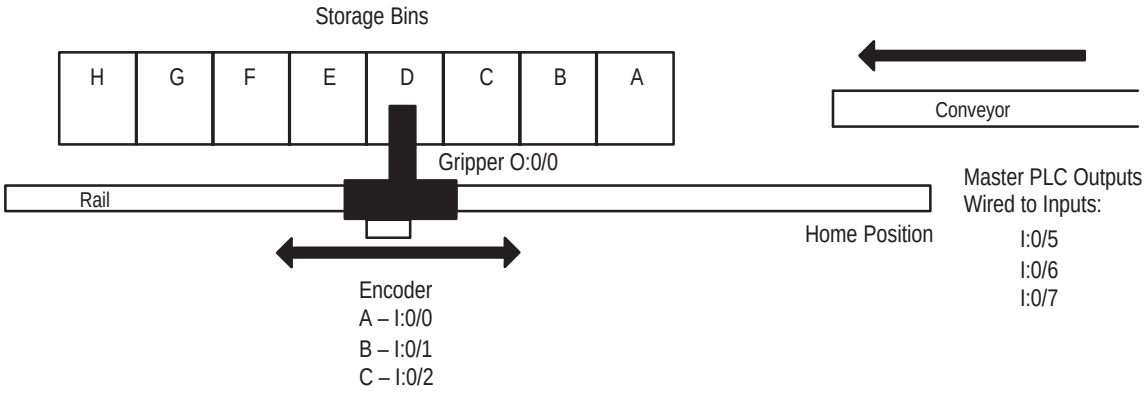


Data Table
Addresses Data (Radix=Decimal)
N7:0 1 1 350 0 0

Pick and Place Machine Example

The following application example illustrates how the controller high-speed counter is configured for the up and down counter using an encoder with reset and hold. For a detailed explanation of:

- XIC, XIO, OTE, RES, OTU, OTL, and TON instructions, see chapter 6.
- GRT and NEQ instructions, see chapter 7.
- MOV instruction, see chapter 9.
- HSC and HSL instructions, see chapter 12.



Pick and Place Machine Operation Overview

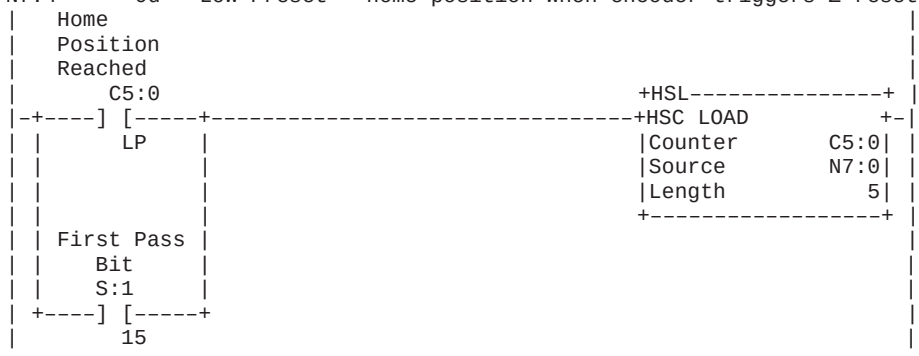
A pick and place machine takes parts from a conveyor and drops them into the appropriate bins. When the pick and place head is positioned over the conveyor with a gripped part, the master PLC communicates to the controller controlling the gripper which bin to drop the part into. This information is communicated by energizing three outputs that are wired to the controller's inputs. Once the controller has this information, it grabs the part and moves down the rail. When the gripper reaches the appropriate bin, it opens and the part falls into the bin. The gripper then returns to the conveyor to retrieve another part.

The position of the pick and place head is read by the controller via a 1000 line quadrature encoder wired to the controller's high-speed counter inputs. When the gripper is in the home position, the Z pulse from the encoder resets the high-speed counter. The number of pulses the head needs to travel to reach each bin location is stored in a data table starting at address N7:10 and ending at N7:17. The controller uses indexed addressing to locate the correct encoder count from the data table and load the information into the high preset of the high-speed counter.

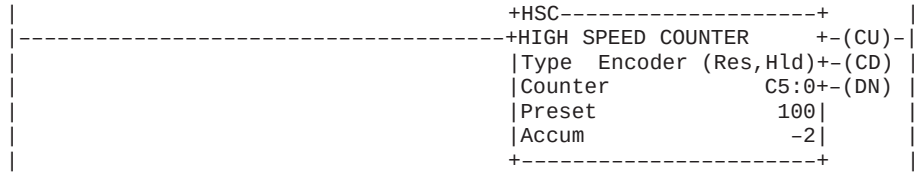
Pick and Place Machine Ladder Program

```
Rung 2:0
The following 3 rungs take information from the other programmable
controller and load it into the INDEX REGISTER.  This will be used to
select the proper bin location from the table starting at N7:10.
| Output    |                                                          |
| from      |                                                          |
| barcode   |                                                Index Reg |
|    I:0    |                                                    S:24  |
|----] [-------------------------------------------------------( )-----|
|       5                                                        0    |
Rung 2:1
| Output    |                                                          |
| from      |                                                          |
| barcode   |                                                Index Reg |
|    I:0    |                                                    S:24  |
|----] [-------------------------------------------------------( )-----|
|       6                                                        1    |
Rung 2:2
| Output    |                                                          |
| from      |                                                          |
| barcode   |                                                Index Reg |
|    I:0    |                                                    S:24  |
|----] [-------------------------------------------------------( )-----|
|       7                                                        2    |
Rung 2:3
Indexes into the table of bin locations and places the correct number
of encoder counts into the high preset of the high-speed counter.
|                                              +MOV---------------+ |
|----------------------------------------------+MOVE               +-|
|                                              |Source      #N7:10| |
|                                              |                100| |
|                                              |Dest          N7:2| |
|                                              |                100| |
|                                              +------------------+ |
```

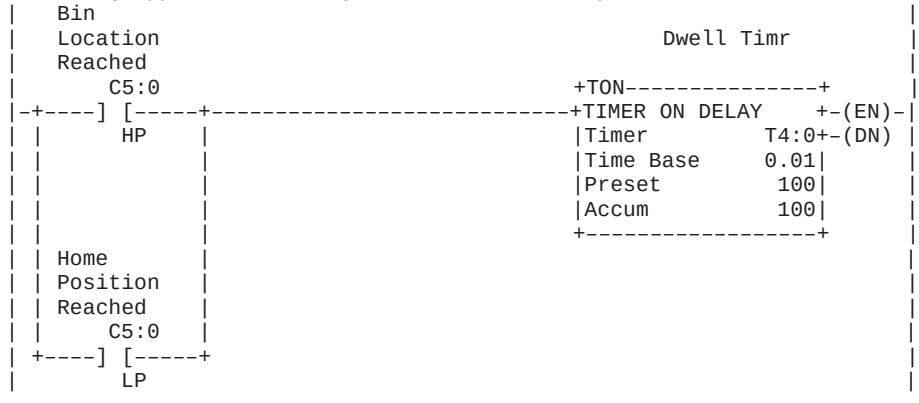
Rung 2:4
Loads the high-speed counter with the following parameters:
N7:0 – 0001h – Output Mask – high-speed counter control only O:0/0 (gripper)
N7:1 – 0000h – Output Pattern for High Preset – turn OFF gripper (release part)
N7:2 – 100d – High Preset – loaded from table in the rung above
N7:3 – 0001h – Output Pattern for Low Preset – turn ON gripper (grab part)
N7:4 – 0d – Low Preset – home position when encoder triggers Z-reset



Rung 2:5
Start up the high-speed counter with the above parameters. Each time this rung is evaluated the hardware accumulator is written to C5:0.ACC.



Rung 2:6
When the pick and place head reaches either its home position to pick up a part or its destination bin to drop off a part, start up a dwell timer. The purpose of this is to keep the head stationary long enough for the gripper to either grab or release the part.



```
Rung 2:7
When the pick and place head is positioned over the proper bin, turn
off the forward motor.  At the same time the high-speed counter will
tell the gripper to release the part and start the dwell timer.  After
the dwell time has expired, start up the reverse motor to send the head
back to its home position to pick up another part.
| Bin                                                        Motor        |
| Location                                                   FORWARD      |
| Reached                                                                 |
|     C5:0                                                      O:0       |
|----] [-------------------------------------+---------------(U)-----+-|
|      HP                                    |                  1    | |
|                                            | Dwell     |Motor      | |
|                                            | Done      |REVERSE    | |
|                                            |     T4:0      O:0     | |
|                                            +----] [--------(L)-----+ |
|                                                   DN          2     |
Rung 2:8
When the pick and place head is positioned at its home position, turn
off the reverse  motor.  At the same time the high-speed counter will
tell the gripper to grab the next part and start the dwell timer.
After the dwell time has expired, start up the forward motor to send
the head out to its drop off bin.
| Home                                                       Motor        |
| Position                                                   REVERSE      |
| Reached                                                                 |
|     C5:0                                                      O:0       |
|----] [-------------------------------------+---------------(U)-----+-|
|      LP                                    |                  2    | |
|                                            | Dwell     |Motor      | |
|                                            | Done      |FORWARD    | |
|                                            |     T4:0      O:0     | |
|                                            +----] [--------(L)-----+ |
|                                                   DN          1     |
Rung 2:9
|                                                                       |
|---------------------------------+END+---------------------------------|
|                                                                       |
```

Data Table

```
Addresses  Data    (Radix=Decimal)
N7:0    1   0   100 1   0   0   0   0   0   0
N7:10   100 200 300 400 500 600 700 800 0   0
```

RPM Calculation Application Example

The following application example illustrates how to calculate the frequency and RPM of a device (such as an encoder) connected to a high-speed counter. *The calculated values are only valid when counting up.* For a detailed explanation of:

- XIC, XIO, CTU and TON instructions, see chapter 6.
- LES instruction, see chapter 7.
- CLR, MUL, DIV, DDV, ADD, and SUB instructions, see chapter 8.
- MOV instruction, see chapter 9.

RPM Calculation Operation Overview

This is done by manipulating the number of counts that have occurred in the high-speed counter accumulator (C0.ACC) over time. To determine this you must provide the following application specific information:

- N7:2 – Counts per Revolution. (i.e., the number of encoder pulses per revolution i.e., the number of pulses until reset). This value is entered in whole counts. For example, you would enter the value 1000 into N7:2 for a 1000 count A/B/Z encoder.
- T4:0.PRE– The Rate Measurement Period (i.e., the amount of time in which to sample the accumulation of counts). This value is entered in .01 second intervals. For example, enter the value 10 into T4:0.PRE for a .1 second rate measurement period . For an accurate frequency and RPM calculation to occur, the value entered must divide evenly into 100. For example valid=20,10,5,4,2,1 and invalid=11,9,8,7,6,3.

Once you have entered these 2 values the following information is provided:

- N7:1 – Counts per last Rate Measurement Period. This value is updated each end of Rate Measurement Period with the number of counts that have elapsed. Use this value if your application requires high-speed calculations such as velocity.
- N7:4 – Frequency. This value is updated once per second with the number of pulses that occurred in the last second. This value (frequency) is calculated:

$$\text{Frequency (Hz)} = \frac{\text{\# pulses}}{\text{1 second}}$$

- N7:5 – RPM. This value is calculated once per second using the frequency value N7:4 together with the counts per revolution value N7:2. For example, if N7:4 contained the value 2000 (indicates 2000 Hz) and you had specified a 1000 count encoder in N7:2, the RPM calculation for N7:5 would be 120. This equates to 2 encoder revolutions per second. Refer to the following calculation:

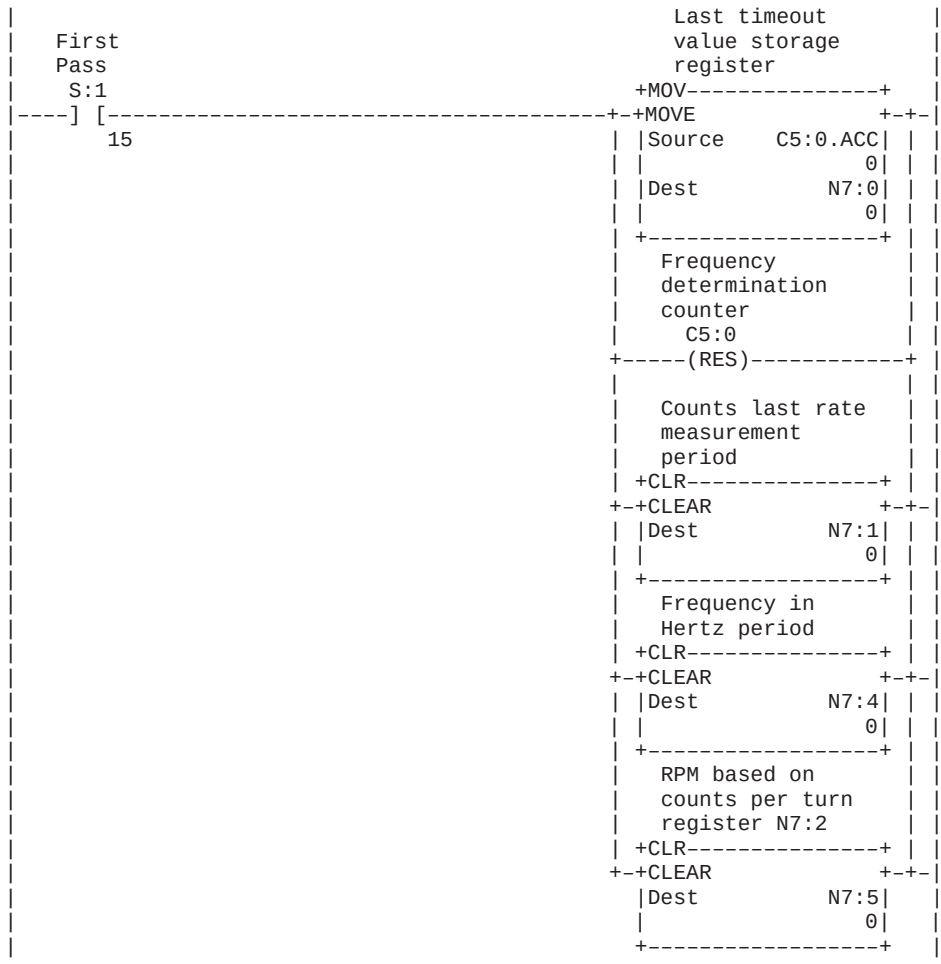
$$\text{RPM} = \frac{\text{\# pulses}}{\text{1 second}} \times \frac{\text{1 revolution}}{\text{\# pulses}} \times \frac{\text{60 seconds}}{\text{1 minute}}$$

$$\text{120 RPM} = \frac{\text{2000 pulses}}{\text{1 second}} \times \frac{\text{1 revolution}}{\text{1000 pulses}} \times \frac{\text{60 seconds}}{\text{1 minute}}$$

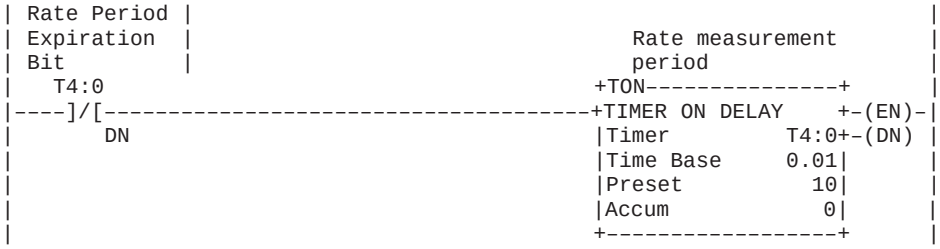
To maintain validity, you must ensure that you cannot accumulate more pulses per rate period than counts per revolution. For example, if you have selected a 1000 pulse encoder, you cannot have more than 999 counts occur in any 1 rate measurement period. If you determine that you exceed this rule, simply lower your Rate Measurement Period T4:0.PRE.

RPM Calculation Ladder Program

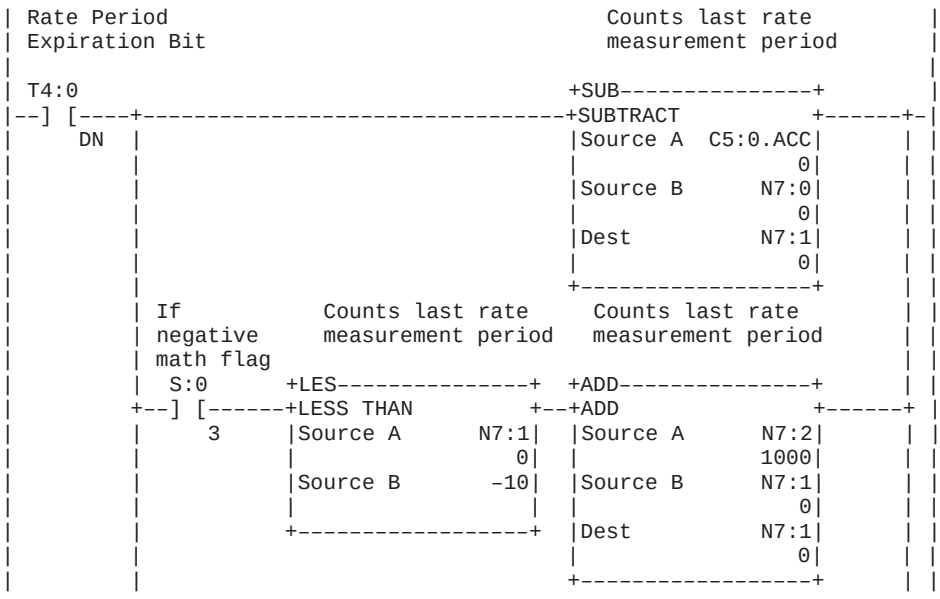
Rung 2:0
Ensures that the measurement value is initialized each REM Run mode entry.

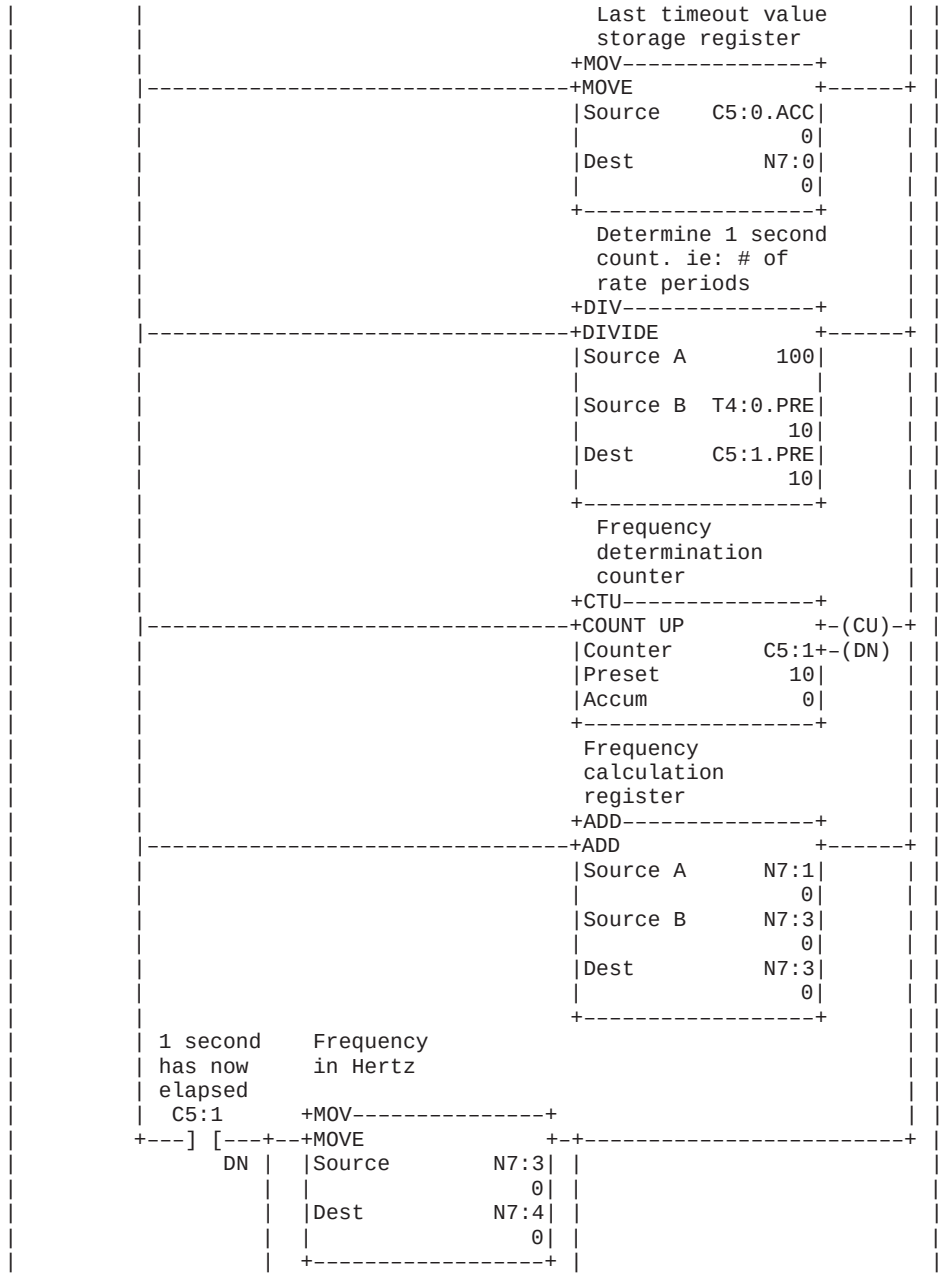


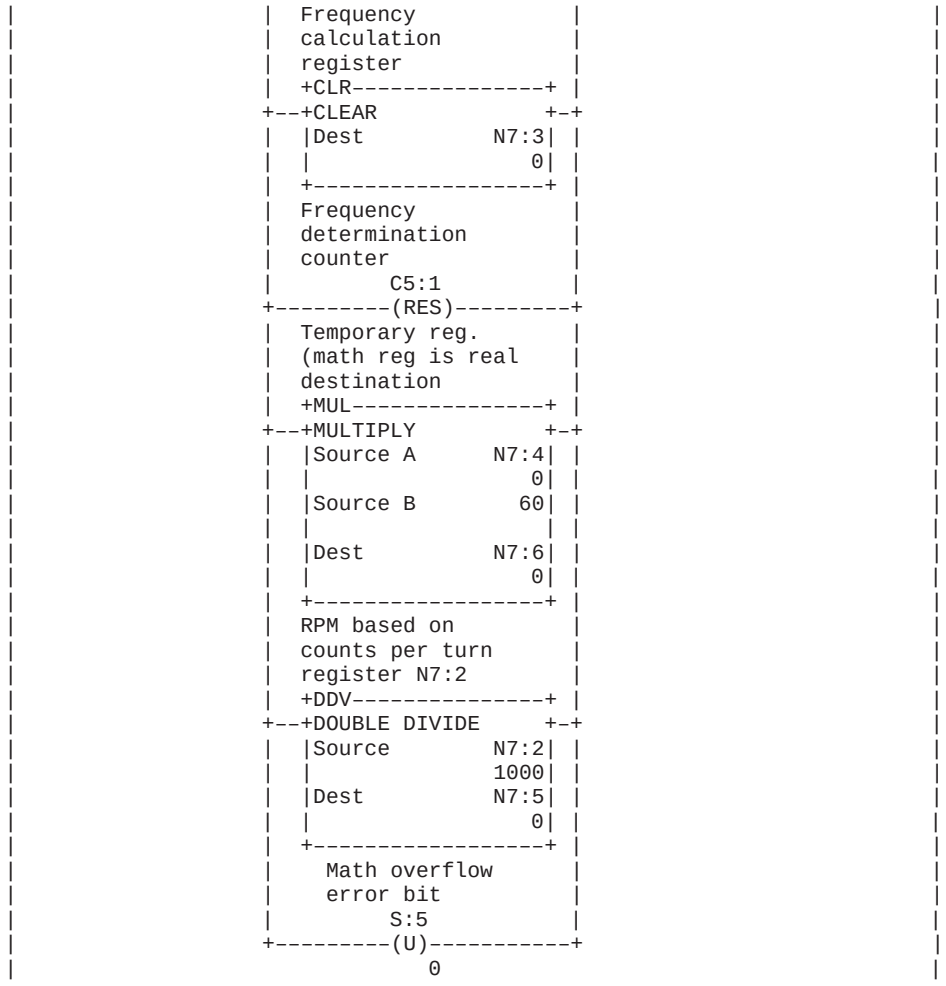
Rung 2:1
Sets the rate measurement period. In this case we are calculating a new rate value once every 100ms. Value N7:1 is updated once every 100ms with the number of counts that have occurred in the last 100ms period. Note that the preset value must divide evenly into 100 in order to accurately determine frequency and RPM (determined later in this program).



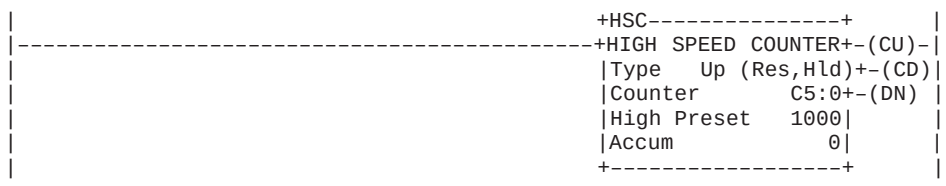
Rung 2:2
Calculates and stores the number of counts that have occurred since the last time that it was executed as true in N7:1 (last time=last rate measurement timer (T4:0) expiration). The LES instruction allows for 10 counts of backlash to occur (you can adjust as needed). The add instruction is configured for a 1000 count encoder using N7:2. (Change this register to match the number of counts generated each Z reset.)



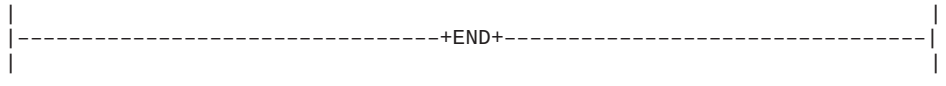

Last timeout value storage register
MOV
MOVE
Source C5:0.ACC
0
Dest N7:0
0
Determine 1 second count. ie: # of rate periods
DIV
DIVIDE
Source A 100
Source B T4:0.PRE
10
Dest C5:1.PRE
10
Frequency determination counter
CTU
COUNT UP
Counter C5:1
Preset 10
Accum 0
Frequency calculation register
ADD
ADD
Source A N7:1
0
Source B N7:3
0
Dest N7:3
0
1 second has now elapsed
C5:1
DN
Frequency in Hertz
MOV
MOVE
Source N7:3
0
Dest N7:4
0



Rung 2:3



Rung 2:4



On/Off Circuit Application Example

The following application example illustrates how to use an input to toggle an output either on or off. For a detailed explanation of:

- XIC, XIO, OTE, OTU, OTL, and OSR instructions, see chapter 6.
- JMP and LBL instructions, see chapter 10.

If the output is off when the input is energized, the output is turned on. If the output is on when the input is energized, the output is turned off.

On/Off Circuit Ladder Program

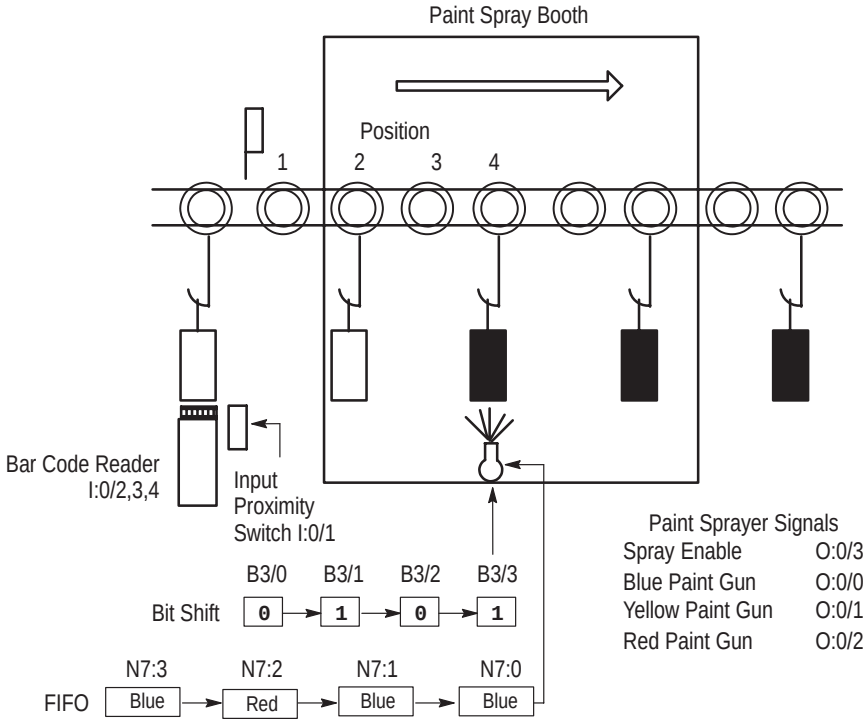
```
Rung 2:0
Does a one-shot from the input push button to an internal bit – the
internal bit is true for only one scan.  This prevent toggling of the
physical output in case the push button is held ”ON” for more than one
scan (always the case).

| push button|OSR #1   |                                        push button |
|    Input  |          |                                        false-to-   |
|                      |                                        true        |
|    I:0        B3                                                B3      |
|----] [-------[OSR]-----------------------------------------( )-----|
|       0          1                                                0    |
Rung 2:1
If the push button input has gone from false-to-true and the output is
presently OFF, turn the output ON and jump over the following rung to
the rest of the programs.  If the JMP instruction was missing, the
following rung would be true and would turn the output back OFF.

|push button|Toggling                                          Toggling     |
| false-to- |Output                                            Output       |
| true      |                                                               |
|    B3         O:0                                              O:0       |
|----] [--------]/[-----------------------------------+----(L)-----+-|
|       0          0                                 |       0    | |
|                                                    | Go to rest | |
|                                                    | of program | |
|                                                    |            | |
|                                                    |     1      | |
|                                                    +---(JMP)----+ |
|                                                                   |
```

```
Rung 2:2
If the push button input has gone from false-to-true and the output is
presently ON, turns the output OFF.

|push button|Toggling  |                                          Toggling   |
| false-to- |Output    |                                          Output     |
| true      |                                                                |
|    B3        O:0                                                  O:0     |
|----] [--------] [-----------------------------------------------(U)-----|
|       0          0                                                 0    |
Rung 2:3
Contains the label corresponding to the jump instruction in rung 1.
The remainder of your actual program would be placed below this rung.

| Go to rest|                                                    Dummy Bit  |
| of program|                                                               |
|           |                                                               |
|     1                                                        B3       |
|---[LBL]-------------------------------------------------------( )-----|
|                                                                 2    |
Rung 2:4

|                                                                        |
|-------------------------------+END+-------------------------------------|
|                                                                        |
```

Spray Booth Application Example

The following application example illustrates the use of bit shift and FIFO instructions in an automated paint spraying operation. For a detailed explanation of:

- XIC and OTE instructions, see chapter 6.
- EQU and LIM instructions, see chapter 7.
- FFU and FFL instructions, see chapter 9.
- BSL instruction, see chapter 11.



Spray Booth Operation Overview

An overhead conveyor with part carriers (hooks) carries parts from a previous operation to the spray booth. Before the part enters the spray booth, 2 items are checked on the conveyor. The first check is for part presence and the second check is for the needed color. This information is stored and accessed later when the part carrier is in the paint spraying area. A proximity switch is used to check for the presence of a part on the carrier and a barcode reader is used to determine color choice. When the part carrier reaches the spraying area, the previously stored information is accessed. If there is a part on the carrier, it is painted according to its bar code and if the carrier is free, paint is not dispensed.

The bit shift and FIFO instructions store the part presence and color information before each carrier enters the spray booth. Both of these instructions place data into their data structures every time a part carrier actuates the shift limit switch.

If the proximity switch senses a part on the carrier, a 1 is shifted into the shift register. If the carrier is free as it passes the shift limit switch, a 0 is shifted into the shift register. The shift register tracks the part carriers approaching the spraying area.

The FIFO does the same type of shifting, except rather than shifting one bit at a time, the FIFO shifts an entire word at a time. Just before the part carrier actuates the SHIFT limit switch, the barcode reader reads the barcode on the part to determine what color the part should be painted. The barcode reader has three outputs that it sets according to what color the part should be. These outputs are:

- wired to the controller as inputs I:0/2, I:0/3, and I:0/4
- combined together to form an integer which is decoded later in the program

This integer is then shifted into the FIFO when the carrier actuates the SHIFT limit switch.

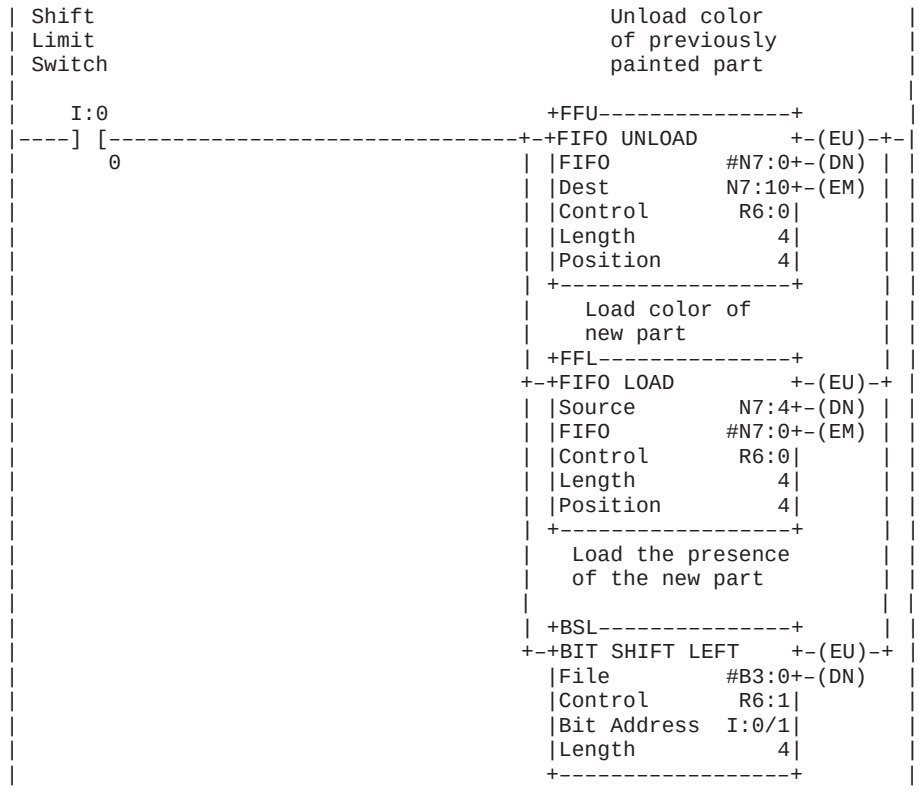
Once the presence and color data is loaded into the shift register and FIFO, they are shifted to new memory locations each time another part carrier actuates the SHIFT limit switch. After three additional shifts, the first part carrier is in front of the spray guns, ready for its part to be painted. At this point the part presence data has been shifted into B3/3 and the color data has been shifted into N7:0. The program now checks B3/3 – if there is a "1" in this location, that means that there is a part hanging on the part carrier and the SPRAY ENABLE output is energized. The program also checks N7:0 to determine which color to paint the part. As the program is checking the shift register for the presence of a part at the spray guns, it is also decoding the color information at N7:0 and energizing the appropriate spray guns. Since we are only using three colors, the only valid color codes are 1, 2, and 3. If any other number is in N7:0 when a part is ready to be painted, the color defaults to BLUE.

Since our program accesses the data while it is still in the two data structures, after the part has been painted, the presence and color information for that part is shifted out of the data structures and lost.

Spray Booth Ladder Program

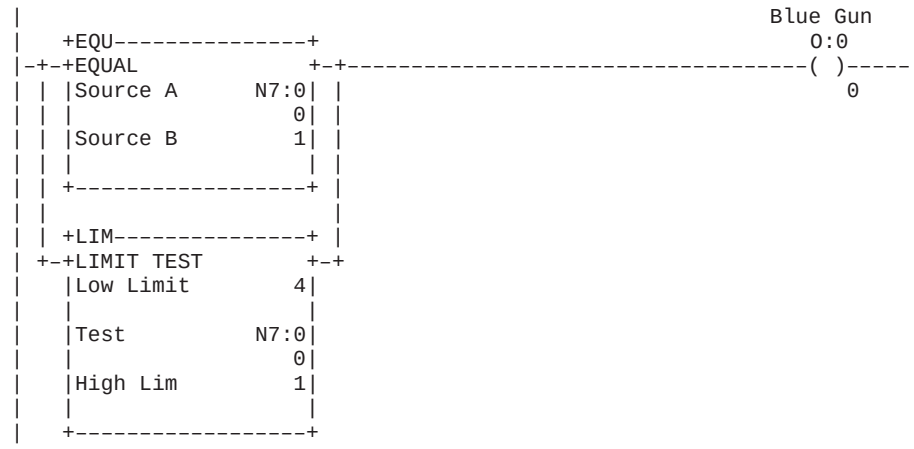
```
Rung 2:0
These three rungs read the color information coming from the barcode
decoder outputs and load this into integer N7:4.  This color is loaded
into the FIFO when the part carrier actuates the SHIFT LIMIT SWITCH.

| Low Bit    |                                                    Color     |
| from Bar   |                                                    Select    |
| Code       |                                                    Word      |
| Decoder    |                                                              |
|    I:0                                                    N7:4      |
|----] [-----------------------------------------------------( )------|
|       2                                                      0     |
Rung 2:1

| Middle Bit |                                                    Color     |
| from Bar   |                                                    Select    |
| Code       |                                                    Word      |
| Decoder    |                                                              |
|    I:0                                                    N7:4      |
|----] [-----------------------------------------------------( )------|
|       3                                                      1     |

Rung 2:2

| High Bit   |                                                    Color     |
| from Bar   |                                                    Select    |
| Code       |                                                    Word      |
| Decoder    |                                                              |
|    I:0                                                     N7:4     |
|----] [-----------------------------------------------------( )-----|
|       4                                                       2    |
```

Rung 2:3
When the part carrier actuates the SHIFT LIMIT SWITCH, three things happen in this rung: (1) the color of the previously painted part is unloaded from the FIFO to make room for the color of the new part, (2) the color of the new part is loaded into the FIFO, (3) the presence or absence of a part on the part carrier is shifted into the Shift Register.

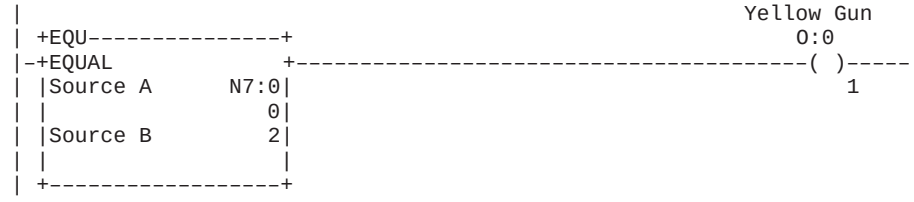


Rung 2:4
If there is a part on the part carrier now entering the spraying area, energize the paint sprayer. If there is not a part on the part carrier, do not energize the sprayer so you can save paint.

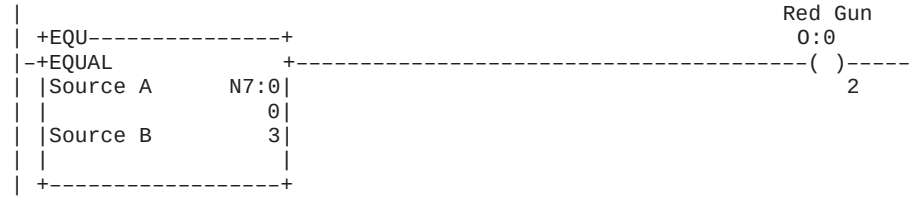
Rung 2:5
Decodes color select word. If N7:0=1 then energize the blue paint gun. Or if N7:0= an invalid color selection, default the color of the part to blue and energize the blue paint gun.



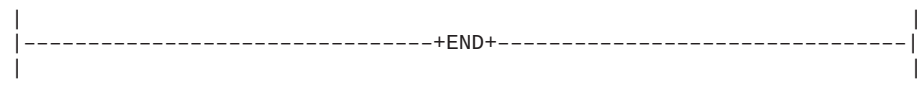
Rung 2:6
Decodes color select word. If N7:0=2 then energize the yellow paint gun.



Rung 2:7
Decodes color select word. If N7:0=3 then energize the red paint gun.



Rung 2:8



Adjustable Timer Application Example

The following application example illustrates the use of timers to adjust the drill dwell time at the end of the machines downstroke. For a detailed explanation of:

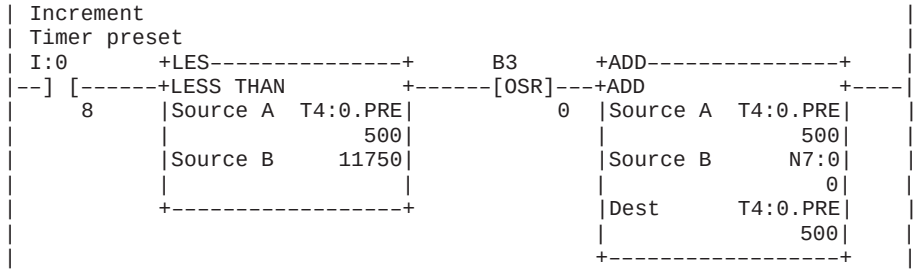
- XIC, TON, and OSR instructions, see chapter 6.
- LES and GRT instructions, see chapter 7.
- ADD and SUB instructions, see chapter 8.

Valid dwell times are 5.0 seconds to 120.0 seconds. Adjustments are made in 2.5 second intervals.

Each time I/8 or I/9 is depressed, the timer preset or delay is adjusted up or down accordingly. By altering the value of N7:0 the amount of change can be increased or decreased. The constants in the LES and GRT instructions, and in the source and destination of the ADD and SUB instructions, could be changed easily to integers for even greater flexibility.

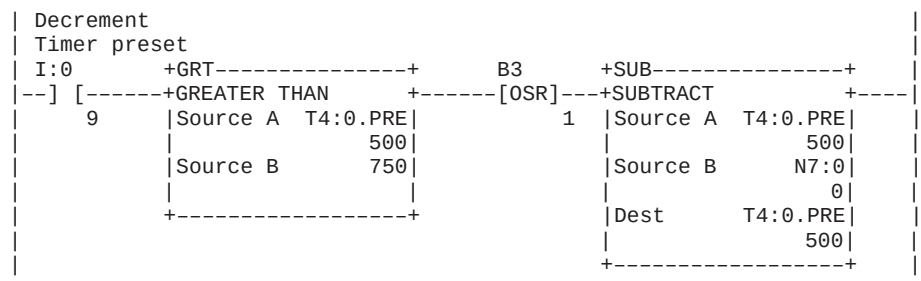
Adjustable Timer Ladder Program

```
Rung 2:0
Adds 2.5 seconds to Timer delay each time the increment push button is
depressed.  Do not exceed 120.0 seconds delay.  Note that N7:0=250.
```

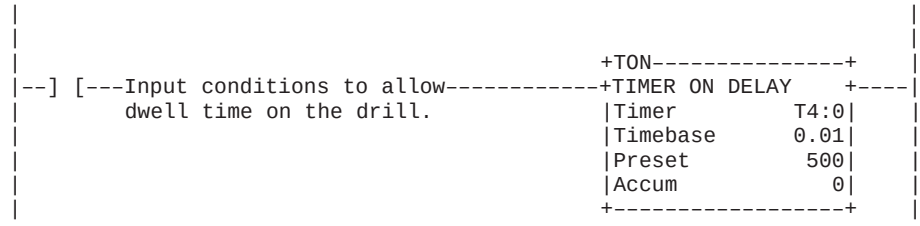


Reference

Rung 2:1
Subtracts 2.5 seconds from Timer delay each time the decrement push button is depressed. Do not go below 5.0 seconds delay.



Rung 2:2



F Optional Analog Input Software Calibration

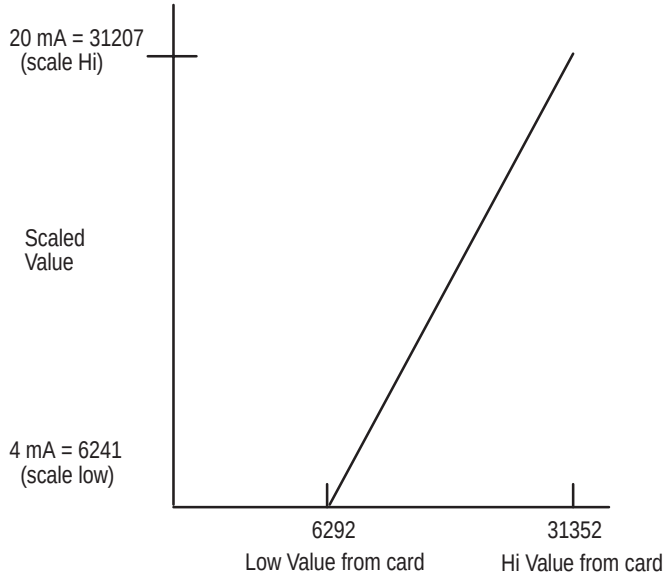
This appendix helps you calibrate an analog input channel using software offsets to increase the expected accuracy of an analog input circuit. Examples of equations and a ladder diagram are provided for your reference. Software calibration reduces the error at a given temperature by scaling the values read at calibration time.

Calibrating an Analog Input Channel

The following procedure can be adapted to all analog inputs; current or voltage. For this example, the 1761-L20BWA-5A with a 4 mA to 20 mA input is used. The overall error for the MicroLogix 1000 is guaranteed to be not more than ± 0.525 at 25°C.

The overall error of ± 0.525% at 20 mA equates to ± 164 LSB of error, or a code range of 31043 to 31371. Any value in this range is returned by an analog input channel at 20 mA. The expected nominal value at 20 mA is 31207. After performing a software calibration, the overall error is reduced to 5 LSB (0.018%), or a code range of 31202 to 31212.

The graph shown below shows the linear relationship between the input value and the resulting scaled value. The values in this graph are from the example program.



Scaled Value vs. Input Value

Calculating the Software Calibration

Use the following equation to perform the software calibration:

Scaled Value = (input value x slope) + offset

Slope = (scaled max. – scaled min.) / (input max. – input min.)

Offset = Scaled min. – (input min. x slope)

Calibration Procedure

1. Heat up / cool down your MicroLogix 1000 system to the temperature in which it will normally be operating.

2. Determine the scaled high and low values you wish to use in your application. In this example, scaled high value (which corresponds to 20 mA) is 31207 and scaled low value (which corresponds to 4 mA) is 6241.

3. Using an analog calibration source connected to the analog input channel or your system's input device placed at the 4 mA position, capture the low value by setting and then resetting the CAL_LO_ENABLE bit. Ensure that your low value lies within the conversion range of your analog input.

4. Using an analog calibration source connected to the analog input channel or your system's input device placed at the 20 mA position, capture the high value by setting and then resetting the CAL_HI_ENABLE bit. Ensure that your high value lies within the conversion range of your analog input.

5. Set and then reset the CALIBRATE bit. This causes the MicroLogix to calculate the slope and offset values used to perform the error correction to the analog input.

The analog channel is now calibrated to ± 5 LSB at the calibration temperature.

The recommended calibration period is once every 6 months. If an application has a wide range of operating temperatures, a software calibration should be performed every 3 to 4 months.

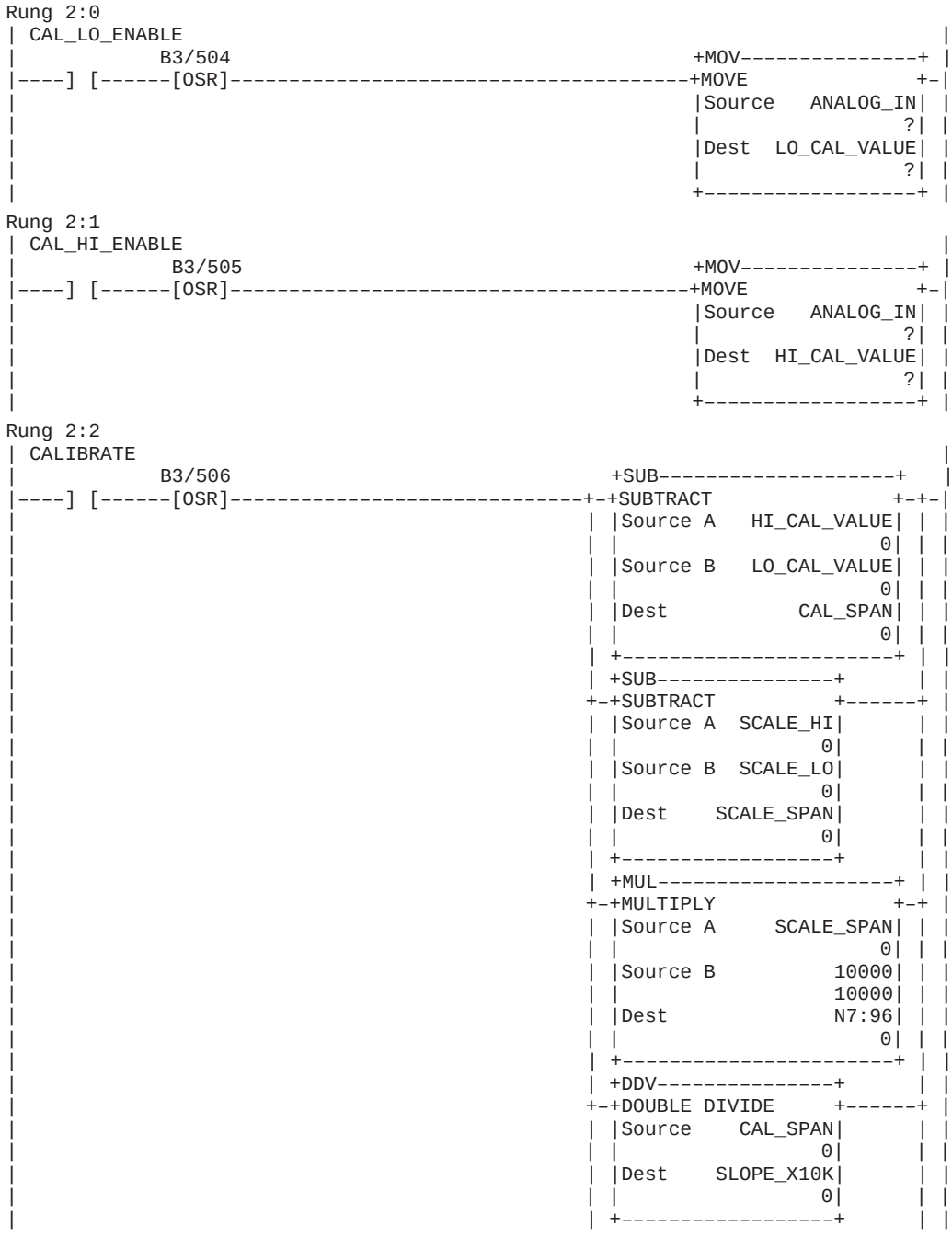
Example Ladder Diagram

The following ladder diagram uses 3 internal bits to perform the calibration procedure. CAL_LO_ENABLE causes the ladder to capture the 4 mA calibration value and CAL_HI_ENABLE causes the ladder to capture the 20 mA calibration value. CALIBRATE causes the ladder diagram to scale the hi and low values to the nominal values, which provides the slope and offset values used to calibrate the analog input channel.

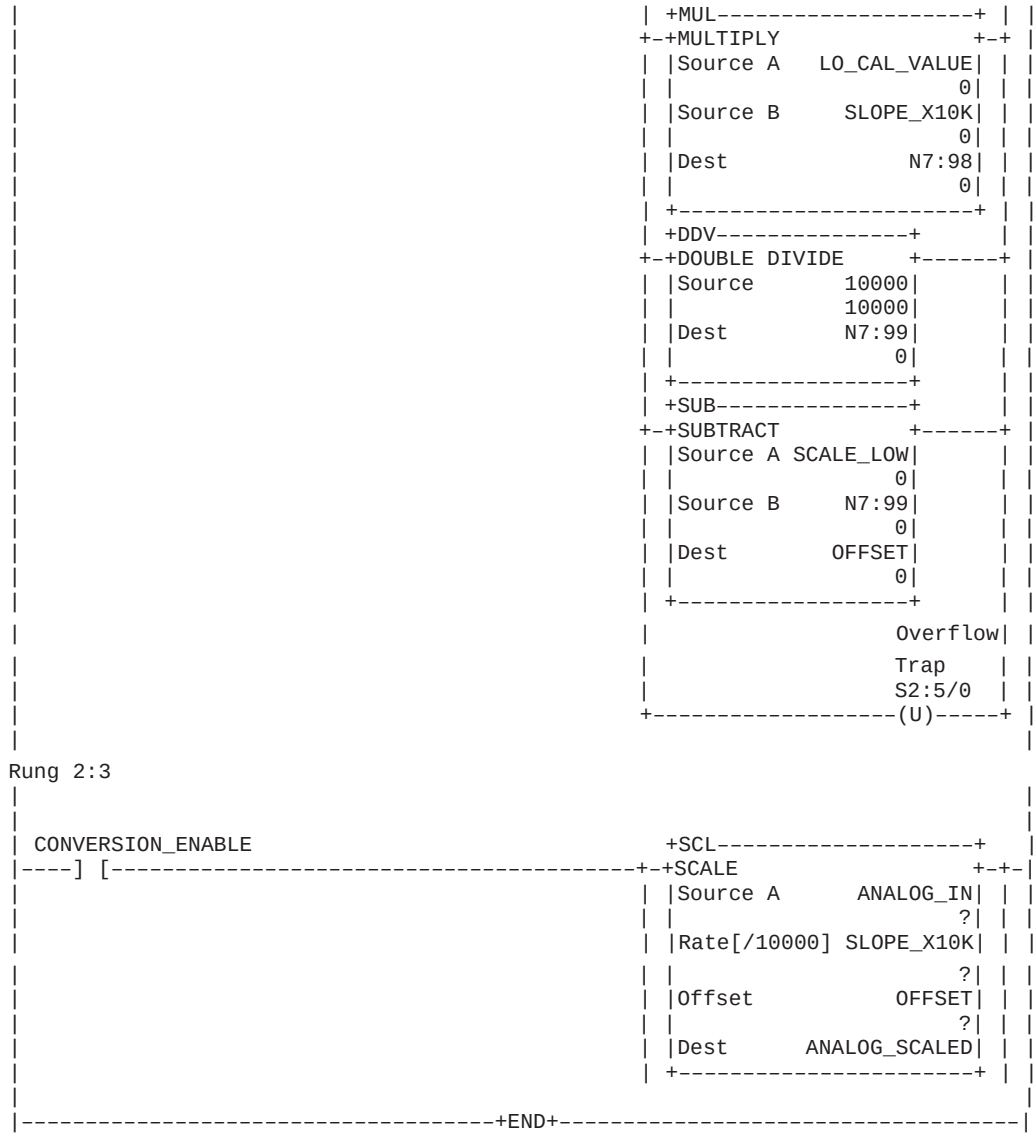
Once the calibration procedure is complete, set the CONVERSION ENABLE bit to a “1”. The calibration numbers can then be used to scale the raw analog data. The corrected analog input data will be placed in memory location ANALOG_SCALED.

The following symbols are used in this example:

CAL_LO_ENABLE	= B3/500
CAL_HI_ENABLE	= B3/501
CALIBRATE	= B3/502
CONVERSION ENABLE	= B3/503
ANALOG_IN	= I:0.4
LO_CAL_VALUE	= N7:90
HI_CAL_VALUE	= N7:91
CAL_SPAN	= N7:92
SCALE_HI	= N7:93
SCALE_LOW	= N7:94
SCALE_SPAN	= N7:95
SLOPE_X10K	= N7:97
OFFSET	= N7:100
ANALOG_SCALED	= N7:101



Ladder logic continued on the next page.


+MUL
+MULTIPLY
Source A LO_CAL_VALUE
0
Source B SLOPE_X10K
0
Dest N7:98
0
+DDV
+DOUBLE DIVIDE
Source 10000
10000
Dest N7:99
0
+SUB
+SUBTRACT
Source A SCALE_LOW
0
Source B N7:99
0
Dest OFFSET
0
Overflow
Trap
S2:5/0
(U)
Rung 2:3
CONVERSION_ENABLE
+SCL
+SCALE
Source A ANALOG_IN
?
Rate[/10000] SLOPE_X10K
?
Offset OFFSET
?
Dest ANALOG_SCALED
+END+

Glossary

The following terms are used throughout this manual. Refer to the *Allen-Bradley Industrial Automation Glossary*, Publication Number AG-7.1, for a complete guide to Allen-Bradley technical terms.

address: A character string that uniquely identifies a memory location. For example, I:1/0 is the memory address for the data located in the Input file location word1, bit 0.

AIC+ Advanced Interface Converter: a device that provides a communication link between various networked devices. (Catalog Number 1761-NET-AIC.)

application: 1) A machine or process monitored and controlled by a controller. 2) The use of computer- or processor-based routines for specific purposes.

backup data: Data downloaded with the program.

baud rate: The speed of communication between devices on a network. All devices must communicate at the same baud rate.

bit: The smallest storage location in memory that contains either a 1 (ON) or a 0 (OFF).

block diagrams: A schematic drawing.

Boolean operators: Logical operators such as AND, OR, NAND, NOR, NOT, and Exclusive-OR that can be used singularly or in combination to form logic statements or circuits. Can have an output response be true or false.

branch: A parallel logic path within a rung of a ladder program.

channel: Refers to the analog signals available on the controller's terminal block. Each channel is configured for connection to a voltage or current source input device, and has its own data and diagnostic status words.

communication scan: A part of the controller's operating cycle. Communication with other devices, such as software running on a personal computer, takes place.

controller: A device, such as a programmable controller, used to monitor input devices and control output devices.

controller overhead: An internal portion of the operating cycle used for housekeeping and set-up purposes.

control profile: The means by which a controller determines which outputs turn on under what conditions.

counter: 1) An electro-mechanical relay-type device that counts the occurrence of some event. May be pulses developed from operations such as switch closures, interruptions of light beams, or other discrete events.
2) In controllers a software counter eliminates the need for hardware counters. The software counter can be given a preset count value to count up or down whenever the counted event occurs.

CPU (Central Processing Unit): The decision-making and data storage section of a programmable controller.

data table: The part of the processor memory that contains I/O values and files where data is monitored, manipulated, and changed for control purposes.

DIN rail: Manufactured according to Deutsche Industrie Normenausshus (DIN) standards, a metal railing designed to ease installation and mounting of your controller.

download: Data is transferred from a programming or storage device to another device.

DTE (Data Terminal Equipment): Equipment that is attached to a network to send or receive data, or both.

EMI: Electromagnetic interference.

encoder: 1) A rotary device that transmits position information. 2) A device that transmits a fixed number of pulses for each revolution.

false: The status of an instruction that does not provide a continuous logical path on a ladder rung.

FIFO (First-In–First-Out): The order that data is entered into and retrieved from a file.

file: A collection of information organized into one group.

full-duplex: A bidirectional mode of communication where data may be transmitted and received simultaneously (contrast with half-duplex).

half-duplex: A communication link in which data transmission is limited to one direction at a time.

hard disk: A storage area in a personal computer that may be used to save processor files and reports for future use.

high byte: Bits 8–15 of a word.

input device: A device, such as a push button or a switch, that supplies signals through input circuits to the controller.

inrush current: The temporary surge current produced when a device or circuit is initially energized.

instruction: A mnemonic and data address defining an operation to be performed by the processor. A rung in a program consists of a set of input and output instructions. The input instructions are evaluated by the controller as being true or false. In turn, the controller sets the output instructions to true or false.

instruction set: The set of general purpose instructions available with a given controller.

I/O (Inputs and Outputs): Consists of input and output devices that provide and/or receive data from the controller.

jump: Change in normal sequence of program execution, by executing an instruction that alters the program counter (sometimes called a branch). In ladder programs a JUMP (JMP) instruction causes execution to jump to a labeled rung.

ladder logic: A program written in a format resembling a ladder-like diagram. The program is used by a programmable controller to control devices.

lease significant bit (LSB): The digit (or bit) in a binary word (code) that carries the smallest value of weight. For the analog controllers, 16-bit two's complement binary codes are used in the I/O image in the card.

LED (Light Emitting Diode): Used as status indicator for processor functions and inputs and outputs.

LIFO (Last-In–First-Out): The order that data is entered into and retrieved from a file.

low byte: Bits 0–7 of a word.

logic: A process of solving complex problems through the repeated use of simple functions that can be either true or false. General term for digital circuits and programmed instructions to perform required decision making and computational functions.

Master Control Relay (MCR): A mandatory hardwired relay that can be de-energized by any series-connected emergency stop switch. Whenever the MCR is de-energized, its contacts open to de-energize all application I/O devices.

mnemonic: A simple and easy to remember term that is used to represent a complex or lengthy set of information.

modem: **Mo**dulator/**dem**odulator. Equipment that connects data terminal equipment to a communication line.

modes: Selected methods of operation. Example: run, test, or program.

negative logic: The use of binary logic in such a way that "0" represents the voltage level normally associated with logic 1 (for example, 0 = +5V, 1 = 0V). Positive is more conventional (for example, 1 = +5V, 0 = 0V).

network: A series of stations (nodes) connected by some type of communication medium. A network may be made up of a single link or multiple links.

nominal input current: The current at nominal input voltage.

normally closed: Contacts on a relay or switch that are closed when the relay is de-energized or the switch is de-activated; they are open when the relay is energized or the switch is activated. In ladder programming, a symbol that will allow logic continuity (flow) if the referenced input is logic "0" when evaluated.

normally open: Contacts on a relay or switch that are open when the relay is de-energized or the switch is de-activated. (They are closed when the relay is energized or the switch is activated.) In ladder programming, a symbol that will allow logic continuity (flow) if the referenced input is logic "1" when evaluated.

offset: The steady-state deviation of a controlled variable from a fixed point.

offline: Describes devices not under direct communication. For example, when programming in APS.

one-shot: A programming technique that sets a bit for only one program scan.

online: Describes devices under direct communication. For example, when APS is monitoring the program file in a controller.

operating voltage: For inputs, the voltage range needed for the input to be in the On state. For outputs, the allowable range of user-supplied voltage.

output device: A device, such as a pilot light or a motor starter coil, that receives data from the controller.

overall accuracy: The worst case deviation of the output voltage or current from the ideal over the full output range is the overall accuracy. For inputs, the worst case deviation of the digital representation of the input signal from the ideal over the full input range is the overall accuracy. This is expressed in percent of full scale.

processor: A Central Processing Unit. (See CPU.)

processor file: The set of program and data files used by the controller to control output devices. Only one processor file may be stored in the controller at a time.

program file: The area within a processor file that contains the ladder logic program.

program mode: When the controller is not executing the processor file and all outputs are de-energized.

program scan: A part of the controller's operating cycle. During the scan the ladder program is executed and the Output data file is updated based on the program and the Input data file.

programming device: Executable programming package used to develop ladder diagrams.

protocol: The packaging of information that is transmitted across a network.

read: To acquire data from a storage place. For example, the processor READs information from the input data file to solve the ladder program.

relay: An electrically operated device that mechanically switches electrical circuits.

relay logic: A representation of the program or other logic in a form normally used for relays.

REM Run mode: REMote run mode during which the processor scans or executes the ladder program, monitors input devices, energizes output devices, and acts on enabled I/O forces.

restore: To download (transfer) a program from a personal computer to a controller.

reserved bit: A status file location that the user should not read or write to.

retentive data: Information associated with data files (timers, counters, inputs, and outputs) in a program that is preserved through power cycles. Program files 2–15 are not effected by retentive data.

RS-232: An EIA standard that specifies electrical, mechanical, and functional characteristics for serial binary communication circuits. A single-ended serial communication interface.

run mode: When the processor file in the controller is being executed, inputs are read, the program is scanned, and outputs are energized and de-energized.

rung: Ladder logic is comprised of a set of rungs. A rung contains input and output instructions. During Run mode, the inputs on a rung are evaluated to be true or false. If a path of true logic exists, the outputs are made true. If all paths are false, the outputs are made false.

save: To upload (transfer) a program stored in memory from a controller to a personal computer; OR to save a program to a computer hard disk.

scan time: The time required for the controller to execute the instructions in the program. The scan time may vary depending on the instructions and each instruction's status during the scan.

sinking: A term used to describe current flow between an I/O device and controller I/O circuit — typically, a sinking device or circuit provides a path to ground, low, or negative side of power supply.

sourcing: A term used to describe current flow between an I/O device and controller I/O circuit — typically, a sourcing device or circuit provides a path to the source, high, or positive side of power supply.

status: The condition of a circuit or system, represented as logic 0 (OFF) or 1 (ON).

terminal: A point on an I/O module that external I/O devices, such as a push button or pilot light, are wired to.

throughput: The time between when an input turns on and the corresponding output turns on.

true: The status of an instruction that provides a continuous logical path on a ladder rung.

update time: For analog inputs, the time between updates to the meory of the analog controller of the digital value representing the analog input signal.

For analog outputs, the time from the digital code being received at the analog controller to the analog output signal of the digital code being output at the terminals of the ouput channel.

upload: Data is transferred to a programming or storage device from another device.

user interrupt poll: While executing the user program, the controller firmware checks for user interrupts that need servicing.

watchdog timer: A timer that monitors a cyclical process and is cleared at the conclusion of each cycle. If the watchdog runs past its programmed time period, it will cause a fault.

workspace: The main storage available for programs and data and allocated for working storage.

write: To copy data to a storage device. For example, the processor WRITEs the information from the output data file to the output modules.

Numbers

1761-L10BWA
features, 1−3
grounding, 2−2
input voltage range, 2−9
mounting, 1−14
output voltage range, 2−9
preventing excessive heat, 1−13
spacing, 1−14
type, 1−3
wiring, 2−4
wiring diagram, 2−9

1761-L10BWB
features, 1−3
grounding, 2−2
input voltage range, 2−12
mounting, 1−14
output voltage range, 2−12
preventing excessive heat, 1−13
spacing, 1−14
type, 1−3
wiring, 2−4
wiring diagram, 2−12

1761-L16AWA
features, 1−3
grounding, 2−2
input voltage range, 2−7
mounting, 1−14
output voltage range, 2−7
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−7

1761-L16BBB
features, 1−3
grounding, 2−2
input voltage range, 2−16
mounting, 1−14
output voltage range, 2−16
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−16

1761-L16BWA
features, 1−3
grounding, 2−2
input voltage range, 2−10
mounting, 1−14
output voltage range, 2−10
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−10

1761-L16BWB
features, 1−3
grounding, 2−2
input voltage range, 2−13
mounting, 1−14
output voltage range, 2−13
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−13

1761-L20AWA-5A
features, 1−3
input voltage range, 2−18
mounting, 1−14
output voltage range, 2−18
preventing excessive heat, 1−13
spacing, 1−14
type, 1−3
wiring diagram, 2−18

1761-L20BWA-5A
features, 1−3
input voltage range, 2−19
mounting, 1−14
output voltage range, 2−19
preventing excessive heat, 1−13
spacing, 1−14
type, 1−3

wiring diagram, 2−19

1761-L20BWB-5A
features, 1−3
input voltage range, 2−20
mounting, 1−14
output voltage range, 2−20
preventing excessive heat, 1−13
spacing, 1−14
type, 1−3
wiring diagram, 2−20

1761-L32AAA
features, 1−3
grounding, 2−2
input voltage range, 2−15
mounting, 1−14
output voltage range, 2−15
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−15

1761-L32AWA
features, 1−3
grounding, 2−2
input voltage range, 2−8
mounting, 1−14
output voltage range, 2−8
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−8

1761-L32BBB
features, 1−3
grounding, 2−2
input voltage range, 2−17
mounting, 1−14
output voltage range, 2−17
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−17

1761-L32BWA
features, 1−3
grounding, 2−2
input voltage range, 2−11
mounting, 1−14
output voltage range, 2−11
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−11

1761-L32BWB
features, 1−3
grounding, 2−2
input voltage range, 2−14
mounting, 1−14
output voltage range, 2−14
preventing excessive heat, 1−13
spacing, 1−14
troubleshooting, 14−2
type, 1−3
wiring, 2−4
wiring diagram, 2−14

32-bit addition and subtraction, 8−6
example, 8−6
math overflow selection bit S:2/14, 8−6

A

accessing processor files
normal operation, 4−7
power up, 4−8

Add (ADD), 8−4
execution times, 8−4
instruction parameters, C−4
updates to arithmetic status bits, 8−4
valid addressing modes, C−4
valid file types, C−4

ADD, Add, 8−4

addressing
data files, 4−10

indexed, 4−12
logical, 4−10
using mnemonics, 4−12

addressing modes, C−3
direct addressing, C−3
immediate addressing, C−3
indexed addressing, C−3

AIC+
applying power to, 3−15
attaching to the network, 3−16
connecting, 3−9
isolated modem, 3−11
network, 3−10
point-to-point, 3−10
installing, 3−16
recommended user supplied components, 3−14
selecting cable, 3−12

Allen-Bradley, contacting for assistance, P−6, 14−10

Allen-Bradley Support, P−6

analog
I/O configuration, 5−3
I/O image, 5−2
input current range, 2−23
input filter and update times, 5−3
input software calibration, F−1
input voltage range, 2−23
output current range, 2−23
output voltage range, 2−23
voltage and current ranges, 2−23

analog , wiring, 2−21

analog cable recommendation, 2−21

analog channels, wiring, 2−22

analog controllers, 1−3
minimizing electrical noise, 2−21

analog data, converting, 5−5

analog input specifications, A−6

analog output specifications, A−6

And (AND), 9−18
execution times, 9−18
instruction parameters, C−4
updates to arithmetic status bits, 9−18
valid addressing modes, C−4
valid file types, C−4

AND, And, 9−18

application example programs
adjustable timer, E−41
bottle line, E−21
conveyor line, E−24
event driven sequencer, E−19
on/off circuit, E−34
paper drilling machine, E−2
RPM calculation, E−28
spray booth, E−36
time driven sequencer, E−17
using the MSG instruction, 13−12

application specific instructions, 11−2
about, 11−2
bit shift instructions, overview, 11−3
Bit Shift Left (BSL), 11−5
Bit Shift Right (BSR), 11−6
in the paper drilling machine application example, 11−21
Selectable Timed Interrupt (STI) function, overview, 11−15
sequencer instructions, overview, 11−7

applying ladder logic to your schematics, 4−14

automatic protocol switching, 3−17

B

basic instructions, 6−2
about, 6−2
bit instructions, overview, 6−3
counter instructions, overview, 6−15
in the paper drilling machine application example, 6−21
timer instructions, overview, 6−8

baud rate
DF1, B−19

DH−485, B−19
limitations for autoswitching, 3−17

bidirectional counter
operation, 12−11
overview, 12−7

bidirectional counter with quadrature encoder
operation, 12−15
overview, 12−7

bidirectional counter with reset and hold
operation, 12−11
overview, 12−7

bidirectional counter with reset and hold with quadrature encoder
operation, 12−15
overview, 12−7

bit file (B3:), 4−5

bit instructions
Examine if Closed (XIC), 6−4
Examine if Open (XIO), 6−4
One-Shot Rising (OSR), 6−7
Output Energize (OTE), 6−5
Output Latch (OTL), 6−5
Output Unlatch (OTU), 6−5
overview, 6−3

bit shift instructions, overview, 11−3
effects on index register S:24, 11−3

Bit Shift Left (BSL), 11−5
effect on index register S:24, 11−4
entering parameters, 11−3
execution times, 11−5, 11−6
instruction parameters, C−4
using, operation, 11−5
valid file types, C−4

Bit Shift Right (BSR), 11−6
effects on index register S:24, 11−4
entering parameters, 11−3
execution times, 11−5, 11−6
instruction parameters, C−4
using, operation, 11−6
valid addressing modes, C−4
valid file types, C−4

BSL, Bit Shift Left, 11−5

BSR, Bit Shift Right, 11−6

C

cables
planning routes for DH-485 connections, D−17
selection guide for the AIC+, 3−12

calibating an analog input channel, F−2

CE mark, 1−2

channel configuration
DF1 full−duplex, D−3
DF1 half−duplex, D−6

Clear (CLR), 8−11
execution times, 8−11
instruction parameters, C−4
updates to arithmetic status bits, 8−11
valid addressing modes, C−4
valid file types, C−4

clearing faults, 14−6

CLR, Clear, 8−11

Common Techniques Used in this Manual, P−6

communication
DeviceNet, 3−18
establishing with controller, 3−17
types of, 13−2

communication protocols
DF1 full-duplex, D−3
DF1 half-duplex, D−5
DH-485, D−11

comparison instructions, 7−1, 7−2
about, 7−2
Equal (EQU), 7−3
Greater Than (GRT), 7−4
Greater Than or Equal (GEQ), 7−4
Less Than (LES), 7−3
Less Than or Equal (LEQ), 7−4
Limit Test (LIM), 7−6

Masked Comparison for Equal (MEQ), 7−5
Not Equal (NEQ), 7−3
overview, 7−2
indexed word addresses, 7−2

connecting the system, 3−1
AIC+, 3−9
DF1 full-duplex protocol, 3−2
DH-485 network, 3−5

contact protection methods, 1−8

contacting Allen-Bradley for assistance, P−6

contactors (bulletin 100), surge suppressors for, 1−10

Contents of this Manual, P−3

control file (R6:), 4−6

controller
determining faults, 14−2
dimensions, A−9
fault messages, 14−7
features, 1−3
grounding, 2−2
installation, 1−1
mounting, 1−14
mounting template, A−9
operating cycle, 4−3
replacement parts, A−10
spacing, 1−14
specifications, A−2
status file, B−1
troubleshooting, 14−2
types, 1−3, A−2
16 I/O, 1−3
32 I/O, 1−3
wiring
for high-speed counter operation, 2−24
recommendations, 2−4
wire type, 2−4

controller faults, 14−2

controller operation, normal, 14−2

controllers, analog, 1−3

Convert from BCD (FRD), 9−5
example, 9−6
execution times, 9−5
instruction parameters, C−6
updates to arithmetic status bits, 9−5
valid addressing modes, C−6
valid file types, C−6

Convert to BCD (TOD), 9−3
changes to the math register, 9−3
example, 9−4
execution times, 9−3
instruction parameters, C−13
updates to arithmetic status bits, 9−3
valid addressing modes, C−13
valid file types, C−13

converting analog data, 5−5

Converting Analog Input Data, 5−5, 5−6

COP, Copy File, 9−10

Copy File (COP), 9−10
execution times, 9−10
instruction parameters, C−4
using, 9−11
entering parameters, 9−11
valid addressing modes, C−4
valid file types, C−4

Count Down (CTD), 6−19
execution times, 6−19
instruction parameters, C−5
using status bits, 6−19
valid addressing modes, C−5
valid file types, C−5

Count Up (CTU), 6−18
execution times, 6−18
instruction parameters, C−5
using status bits, 6−18
valid addressing modes, C−5
valid file types, C−5

counter file (C5:), 4−6

counter instructions
Count Down (CTD), 6−19
Count Up (CTU), 6−18
in the paper drilling machine application example, 7−8

overview, 6−15
addressing structure, 6−16
entering parameters, 6−16
how counters work, 6−17
Reset (RES), 6−20

CTD, Count Down, 6−19

CTU, Count Up, 6−18

D

data files, 4−5
addressing, 4−10
organization, 4−5
types, 4−10
file indicator (#), 4−13

data handling instructions, 9−2
about, 9−2
Convert from BCD (FRD), 9−5
Convert to BCD (TOD), 9−3
Copy File (COP), 9−10
Decode 4 to 1 of 16 (DCD), 9−8
Encode 1 of 16 to 4 (ENC), 9−9
FIFO and LIFO instructions, overview, 9−23
Fill File (FLL), 9−10
in the paper drilling machine application example, 9−28
move and logical instructions, overview, 9−13

DCD, Decode 4 to 1 of 16, 9−8

DDV, Double Divide, 8−10

Decode 4 to 1 of 16 (DCD), 9−8
entering parameters, 9−8
execution times, 9−8
instruction parameters, C−5
updates to arithmetic status bits, 9−8
valid addressing modes, C−5
valid file types, C−5

developing your logic program−a model, 4−15

DeviceNet Communications, 3−18

DF1 full-duplex protocol
configuration parameters, D−3
connecting, 3−2
description, D−3
example system configuration, D−4
using a modem, 3−3, D−9

DF1 half-duplex protocol
configuration parameters, D−6
description, D−5

DH−485 communication protocol, configuration parameters, D−12

DH-485 network
configuration parameters, D−18
connecting, 3−5
description, D−11
devices that use the network, D−13
example system configuration, D−19
initialization, D−13
installation, 3−5
planning considerations, D−16
protocol, D−11
token rotation, D−12

dimensions, controller, A−9

DIN rail, 1−15
mounting dimensions, 1−15

diode, 1N4004, 1−9

direct addressing, C−3

displaying values, 4−13

DIV, Divide, 8−9

Divide (DIV), 8−9
changes to the math register, 8−9
execution times, 8−9
instruction parameters, C−5
updates to arithmetic status bits, 8−9
valid addressing modes, C−5
valid file types, C−5

Double Divide (DDV), 8−10
changes to the math register, 8−10
execution times, 8−10
instruction parameters, C−5
updates to arithmetic status bits, 8−10
valid addressing modes, C−5

valid file types, C−5

E

Electronics Industries Association (EIA), D−2

EMC Directive, 1−2

emergency-stop switches, 1−5

ENC, Encode 1 of 16 to 4, 9−9

Encode 1 of 16 to 4 (ENC), 9−9
- entering parameters, 9−9
- execution times, 9−9
- instruction parameters, C−5
- updates to arithmetic status bits, 9−10
- valid addressing modes, C−5
- valid file types, C−5

entering
- numeric constants, 4−13
- values, 4−14

EQU, Equal, 7−3

Equal (EQU), 7−3
- execution times, 7−3
- instruction parameters, C−5
- valid addressing modes, C−5
- valid file types, C−5

error recovery model, 14−5

errors, 14−3
- download, B−17
- going−to−run, B−14
- hardware, 14−3
- identifying, 14−6
- MSG instruction, 13−10
- powerup, B−14
- run, B−16, B−17

establishing communication, 3−17

European Union Directive compliance, 1−2

Examine if Closed (XIC), 6−4
- execution times, 6−4
- instruction parameters, C−13
- valid addressing modes, C−13
- valid file types, C−13

Examine if Open (XIO), 6−4
- execution times, 6−4
- instruction parameters, C−13
- valid addressing modes, C−13
- valid file types, C−13

example programs
- adjustable timer, E−41
- bottle line, E−21
- conveyor line, E−24
- event driven sequencer, E−19
- on/off circuit, E−34
- paper drilling machine, E−2
- RPM calculation, E−28
- spray booth, E−36
- time driven sequencer, E−17
- using the MSG instruction, 13−12

Exclusive Or (XOR), 9−20
- execution times, 9−20
- instruction parameters, C−13
- updates to arithmetic status bits, 9−20
- valid addressing modes, C−13
- valid file types, C−13

execution times
- listing, B−1
- worksheet, B−26

F

fault messages, 14−7

fault recovery procedure, 14−6

fault routine, 14−6

FFL, FIFO Load, 9−25

FFU, FIFO Unload, 9−25

FIFO and LIFO instructions
- FIFO Load (FFL), 9−25
- FIFO Unload (FFU), 9−25
- LIFO Load (LFL), 9−26
- LIFO Unload (LFU), 9−26
- overview, 9−23
  - effects on index register S:24, 9−24

entering parameters, 9−23

FIFO Load (FFL), 9−25
execution times, 9−25
instruction parameters, C−6
operation, 9−25
valid addressing modes, C−6
valid file types, C−6

FIFO Unload (FFU), 9−25
execution times, 9−26
instruction parameters, C−6
operation, 9−25
valid addressing modes, C−6
valid file types, C−6

file indicator (#), 4−13

file organization
data files, 4−5
program files, 4−4

file types, C−2

Fill File (FLL), 9−10
execution times, 9−10
instruction parameters, C−6
using, 9−12
entering parameters, 9−12
valid addressing modes, C−6
valid file types, C−6

filter, input, 5−3

filter response times, A−7

filtering, input channel, 5−4

FLL, Fill File, 9−10

FRD, Convert from BCD, 9−5

G

general specifications, A−3

GEQ, Greater Than or Equal, 7−4

Greater Than (GRT), 7−4
execution times, 7−4
instruction parameters, C−6
valid addressing modes, C−6
valid file types, C−6

Greater Than or Equal (GEQ), 7−4
execution times, 7−4
instruction parameters, C−6
valid addressing modes, C−6
valid file types, C−6

grounding the controller, 2−2

GRT, Greater Than, 7−4

H

hardware, features, 1−3

heat protection, 1−13

high-speed counter, wiring, 2−24, 12−7

High-Speed Counter (HSC), 12−6
entering parameters, 12−6
execution times, 12−6
instruction parameters, C−7
types of, 12−7
bidirectional counter, 12−10
bidirectional counter with reset and hold, 12−10
bidirectional counter with reset and hold with a quadrature encoder, 12−14
up counter, 12−8
up counter with reset and hold, 12−8
valid addressing modes, C−7
valid file types, C−7
what happens when going to REM Run, 12−25

high-speed counter instructions, 12−2
about, 12−2
High-Speed Counter (HSC), 12−6
High-Speed Counter Interrupt Disable (HSD), 12−23
High-Speed Counter Interrupt Enable (HSE), 12−23
High-Speed Counter Load (HSL), 12−18
High-Speed Counter Reset Accumulator (RAC), 12−22
in the paper drilling machine application example, 12−29
overview, 12−3

High-Speed Counter Interrupt Disable (HSD), 12−23
  execution times, 12−23
  instruction parameters, C−7
  using HSD, 12−24
    operation, 12−24
  valid addressing modes, C−7
  valid file types, C−7

High-Speed Counter Interrupt Enable (HSE), 12−23
  execution times, 12−23
  instruction parameters, C−7
  using HSE, 12−23
    operation, 12−23
  valid addressing modes, C−7
  valid file types, C−7

High-Speed Counter Load (HSL), 12−18
  entering parameters, 12−18
  execution times, 12−18
  instruction parameters, C−7
  operation, 12−18
  valid addressing modes, C−7
  valid file types, C−7

High-Speed Counter Reset Accumulator (RAC), 12−22
  entering parameters, 12−22
  execution times, 12−22
  instruction parameters, C−10
  operation, 12−22
  valid addressing modes, C−10
  valid file types, C−10

HSC, High-Speed Counter, 12−6

HSD, High-Speed Counter Interrupt Disable, 12−23

HSE, High-Speed Counter Interrupt Enable, 12−23

HSL, High-Speed Counter Load, 12−18

I

I/O configuration, analog, 5−3

I/O image, analog, 5−2

identifying controller faults, 14−6

IIM, Immediate Input with Mask, 10−9

Immediate Input with Mask (IIM), 10−9
  entering parameters, 10−9
  execution times, 10−9
  instruction parameters, C−7
  valid addressing modes, C−7
  valid file types, C−7

Immediate Output with Mask (IOM), 10−9
  entering parameters, 10−9
  execution times, 10−9
  instruction parameters, C−8
  valid addressing modes, C−8
  valid file types, C−8

indexed addressing, 4−12, C−3
  example, 4−12
  specifying, 4−12

Input Channel Filtering, 5−4

input current range, analog, 2−23

input file (I:), 4−5

Input Filter, analog, 5−3

input filter settings, A−7

input specifications, A−4

Input States on Power Down, 1−13

input voltage ranges
  1761-L10BWA, 2−9
  1761-L10BWB, 2−12
  1761-L16AWA, 2−7
  1761-L16BBB, 2−16
  1761-L16BWA, 2−10
  1761-L16BWB, 2−13
  1761-L20AWA-5A, 2−18
  1761-L20BWA-5A, 2−19
  1761-L20BWB-5A, 2−20
  1761-L32AAA, 2−15
  1761-L32AWA, 2−8
  1761-L32BBB, 2−17
  1761-L32BWA, 2−11
  1761-L32BWB, 2−14
  analog, 2−23

installing, the micro controller, 1−1

instruction execution time, worksheet, B−26

instruction execution times, listing, B−21

instruction memory usage
  listing, B−21
  worksheet, B−25

instruction set, C−1

INT, Interrupt Subroutine, 11−20

integer file (N7:), 4−6

interrupt latency
  STI, 11−16
  user, B−24

interrupt priorities, 11−17

Interrupt Subroutine (INT), 11−20
  execution times, 11−20
  instruction parameters, C−7
  valid addressing modes, C−7
  valid file types, C−7

IOM, Immediate Output with Mask, 10−9

isolated link coupler, installing, 3−6

J

JMP, Jump, 10−2

JSR, Jump to Subroutine, 10−4

Jump (JMP), 10−2
  entering parameters, 10−2
  execution times, 10−2
  instruction parameters, C−8
  using, 10−2
  valid addressing modes, C−8
  valid file types, C−8

Jump to Subroutine (JSR), 10−4
  execution times, 10−4
  instruction parameters, C−8
  nesting subroutine files, 10−5
  using, 10−5
  valid addressing modes, C−8
  valid file types, C−8

L

Label (LBL), 10−2
  entering parameters, 10−2
  execution times, 10−2
  instruction parameters, C−8
  using, 10−3
  valid addressing modes, C−8
  valid file types, C−8

ladder logic
  applying to your schematics, 4−14
  developing your logic program, 4−15

LBL, Label, 10−2

LEDs, 14−2
  error with controller, 14−3
  normal controller operation, 14−2

LEQ, Less Than or Equal, 7−4

LES, Less Than, 7−3

Less Than (LES), 7−3
  execution times, 7−3
  instruction parameters, C−8
  valid addressing modes, C−8
  valid file types, C−8

Less Than or Equal (LEQ), 7−4
  execution times, 7−4
  instruction parameters, C−8
  valid addressing modes, C−8
  valid file types, C−8

LFL, LIFO Load, 9−26

LFU, LIFO Unload, 9−26

LIFO Load (LFL), 9−26
  execution times, 9−27
  instruction parameters, C−8
  operation, 9−26
  valid addressing modes, C−8
  valid file types, C−8

LIFO Unload (LFU), 9−26
  execution times, 9−27
  instruction parameters, C−8
  operation, 9−26
  valid addressing modes, C−8

valid file types, C−8

LIM, Limit Test, 7−6

Limit Test (LIM), 7−6
entering parameters, 7−6
execution times, 7−6
instruction parameters, C−9
valid addressing modes, C−9
valid file types, C−9

logical address, 4−10

logical addresses, specifying, using mnemonics, 4−12

M

machine control, principles of, 4−2

manuals, related, P−5

Masked Comparison for Equal (MEQ), 7−5
entering parameters, 7−5
execution times, 7−5
instruction parameters, C−9
valid addressing modes, C−9
valid file types, C−9

Masked Move (MVM), 9−16
entering parameters, 9−16
execution times, 9−16
instruction parameters, C−10
operation, 9−17
updates to arithmetic status bits, 9−16
valid addressing modes, C−10
valid file types, C−10

Master Control Relay, 1−4

Master Control Reset (MCR), 10−7
execution times, 10−7
instruction parameters, C−9
valid addressing modes, C−9
valid file types, C−9

master/sender communication, 13−2

math instructions, 8−2
32-bit addition and subtraction, 8−6
about, 8−2
Add (ADD), 8−4
Clear (CLR), 8−11
Divide (DIV), 8−9
Double Divide (DDV), 8−10
in the paper drilling machine application example, 8−14
Multiply (MUL), 8−8
overview, 8−2
changes to the math register, S:13 and S:14, 8−3
overflow trap bit, S:5/0, 8−3
updates to arithmetic status bits, 8−2
using indexed word addresses, 8−2
Scale Data (SCL), 8−12
Square Root (SQR), 8−11
Subtract (SUB), 8−5
using arithmetic status bits, 9−10

MCR, Master Control Reset, 10−7

MEQ, Masked Comparison for Equal, 7−5

Message (MSG), 13−1
application examples, 13−12
control block layout, 13−5
entering parameters, 13−3
error codes, 13−10
execution times, 13−3
instruction parameters, C−9
timing diagram, 13−8
using status bits, 13−6
valid addressing modes, C−9
valid file types, C−9

mnemonic, using, in logical addresses, 4−12

model for developing a logic program, 4−15

modem cable, constructing your own, 3−11

modems
dial-up phone , D−9
leased-line, D−9
line drivers, D−10
radio, D−10
using with MicroLogix controllers, D−9

monitoring, controller operation, fault recovery procedure, 14−6

motor starters (bulletin 509), surge suppressors, 1−10

motor starters (bulletin 709), surge suppressors, 1−10

mounting template, A−9

mounting the controller
- using a DIN rail, 1−15
- using mounting screws, 1−16
- vertically, 1−16

MOV, Move, 9−15

Move (MOV), 9−15
- entering parameters, 9−15
- execution times, 9−15
- instruction parameters, C−9
- updates to arithmetic status bits, 9−15
- valid addressing modes, C−9
- valid file types, C−9

move and logical instructions
- And (AND), 9−18
- Exclusive Or (XOR), 9−20
- Masked Move (MVM), 9−16
- Move (MOV), 9−15
- Negate (NEG), 9−22
- Not (NOT), 9−21
- Or (OR), 9−19
- overview, 9−13
  - changes to the math register, S:13 and S:14, 9−14
  - entering parameters, 9−13
  - overflow trap bit, S:5/0, 9−14
  - updates to arithmetic status bits, 9−13
  - using indexed word addresses, 9−13

MSG, Message, 13−1

MUL, Multiply, 8−8

Multiply (MUL), 8−8
- changes to the math register, 8−8
- execution times, 8−8
- instruction parameters, C−10
- updates to arithmetic status bits, 8−8
- valid addressing modes, C−10
- valid file types, C−10

MVM, Masked Move, 9−16

N

NEG, Negate, 9−22

Negate (NEG)
- instruction parameters, C−10
- valid addressing modes, C−10
- valid file types, C−10

Negate (NEG), 9−22
- execution times, 9−22
- updates to arithmetic status bits, 9−22

NEQ, Not Equal, 7−3

nesting subroutine files, 10−5

node address (S:15L), B−18, B−19

nominal transfer function, 5−5

Not (NOT), 9−21
- execution times, 9−21
- instruction parameters, C−10
- updates to arithmetic status bits, 9−21
- valid addressing modes, C−10
- valid file types, C−10

Not Equal (NEQ), 7−3
- execution times, 7−3
- instruction parameters, C−10
- valid addressing modes, C−10
- valid file types, C−10

NOT, Not, 9−21

null modem cable, 3−4

number systems, 4−13
- radices used, 4−13

numeric constants, 4−13

O

One-Shot Rising (OSR), 6−7
- entering parameters, 6−7
- example rung, 6−7
- execution times, 6−7
- instruction parameters, C−10
- valid addressing modes, C−10

valid file types, C−10

operating cycle, controller's, 4−3

Or (OR), 9−19
execution times, 9−19
instruction parameters, C−10
updates to arithmetic status bits, 9−19
valid addressing modes, C−10
valid file types, C−10

OR, Or, 9−19

OSR, One-Shot Rising, 6−7

OTE, Output Energize, 6−5

OTL, Output Latch, 6−5

OTU, Output Unlatch, 6−5

output contact protection, selecting, 1−8

output current range, analog, 2−23

Output Energize (OTE), 6−5
execution times, 6−5, 12−24
instruction parameters, C−10
valid addressing modes, C−10
valid file types, C−10

output file (O:), 4−5

Output Latch (OTL), 6−5
execution times, 6−5
instruction parameters, C−10
using, 6−6
valid addressing modes, C−10
valid file types, C−10

output specifications, A−5

Output Unlatch (OTU), 6−5
execution times, 6−5
instruction parameters, C−10
using, 6−6
valid addressing modes, C−10
valid file types, C−10

output voltage ranges
1761-L10BWA, 2−9
1761-L10BWB, 2−12
1761-L16AWA, 2−7
1761-L16BBB, 2−16
1761-L16BWA, 2−10
1761-L16BWB, 2−13
1761-L20AWA-5A, 2−18
1761-L20BWA-5A, 2−19
1761-L20BWB-5A, 2−20
1761-L32AAA, 2−15
1761-L32AWA, 2−8
1761-L32BBB, 2−17
1761-L32BWA, 2−11
1761-L32BWB, 2−14
analog, 2−23

overflow trap bit, S:5/0, 8−3

overview
bit instructions, 6−3
comparison instructions, 7−2
counter instructions, 6−15
FIFO and LIFO instructions, 9−23
high-speed counter instructions, 12−3
math instructions, 8−2
move and logical instructions, 9−13
Selectable Timed Interrupt (STI) function, 11−15
timer instructions, 6−8

ownership timeout, D−8

P

planning considerations for a network, D−16

Power Considerations
Input States on Power Down, 1−13
Isolation Transformers, 1−12
Loss of Power Source, 1−12
other line conditions, 1−13
overview, 1−12

Power Distribution, 1−11

preventing excessive heat, 1−13

principles of machine control, 4−2

processor files
organization, 4−4
overview, 4−4
data files, 4−5
program files, 4−5

storing and accessing, 4−6
download, 4−7
normal operation, 4−7
power down, 4−8
power up, 4−8
program constants, 4−13
program development model, 4−15
program faults, determining, 14−2
program files, 4−4, 4−5
program flow control instructions, 10−2
about, 10−2
Immediate Input with Mask (IIM), 10−9
Immediate Output with Mask (IOM), 10−9
in the paper drilling machine application example, 10−10
Jump (JMP), 10−2
Jump to Subroutine (JSR), 10−4
Label (LBL), 10−2
Master Control Reset (MCR), 10−7
Return (RET), 10−4
Subroutine (SBR), 10−4
Suspend (SUS), 10−8
Temporary End (TND), 10−8
programming overview, 4−1
protection methods for contacts, 1−8
protocol switching, automatic, 3−17
publications, related, P−5
Purpose of this Manual, P−2

Q

quadrature encoder input, 12−14

R

RAC, High-Speed Counter Reset Accumulator, 12−22
RC network, example, 1−9
related publications, P−5
relay contact rating table, A−5
relays, surge suppressors for, 1−10
remote packet support, D−22
replacement parts, controller, A−10
RES, Reset, 6−20
Reset (RES), 6−20
execution times, 6−20
instruction parameters, C−11
resetting the high−speed counter accumulator
instruction parameters, C−11
valid addressing modes, C−11
valid file types, C−11
resetting the high-speed counter accumulator, 12−21
execution times, 12−21
operation, 12−21
valid addressing modes, C−11
valid file types, C−11
RET, Return, 10−4
Retentive Timer (RTO), 6−14
execution times, 6−14
instruction parameters, C−11
using status bits, 6−14
valid addressing modes, C−11
valid file types, C−11
Return (RET), 10−4
execution times, 10−4
instruction parameters, C−11
nesting subroutine files, 10−5
using, 10−6
valid addressing modes, C−11
valid file types, C−11
RS-232 communication interface, D−2
RTO, Retentive Timer, 6−14

S

Safety Considerations
Disconnecting Main Power, 1−11

overview, 1−11
Periodic Tests of Master Control Relay Circuit, 1−12
Power Distribution, 1−11
Safety Circuits, 1−11

SBR, Subroutine, 10−4

Scale (SCL)
instruction parameters, C−11
valid addressing modes, C−11
valid file types, C−11

Scale Data (SCL), 8−12
application example, 8−13
entering parameters, 8−12
execution times, 8−12
updates to arithmetic status bits, 8−12

SCL, Scale Data, 8−12

Selectable Timed Disable (STD), 11−18
example, 11−18
execution times, 11−18
instruction parameters, C−12
using, 11−18
valid addressing modes, C−12
valid file types, C−12

Selectable Timed Enable (STE), 11−18
example, 11−18
execution times, 11−18
instruction parameters, C−12
using, 11−18
valid addressing modes, C−12
valid file types, C−12

Selectable Timed Interrupt (STI) function
basic programming procedure, 11−15
Interrupt Subroutine (INT), 11−20
operation, 11−15
interrupt latency and interrupt occurrences, 11−16
interrupt priorities, 11−17
status file data saved, 11−17
subroutine content, 11−16
overview, 11−15
Selectable Timed Disable (STD), 11−18
Selectable Timed Enable (STE), 11−18
Selectable Timed Start (STS), 11−20
STD/STE zone example, 11−18

Selectable Timed Start (STS), 11−20
execution times, 11−20
instruction parameters, C−12
valid addressing modes, C−12
valid file types, C−12

selected DF1 protocol bit, S:0/12, B−4

Selecting Surge Suppressors, 1−8

Sequencer Compare (SQC), 11−7
entering parameters, 11−8
execution times, 11−7
instruction parameters, C−11
using, 11−11
valid addressing modes, C−11
valid file types, C−11

sequencer instructions
overview, 11−7
effects on index register S:24, 11−7
Sequencer Compare (SQC), 11−7
Sequencer Load (SQL), 11−13
Sequencer Output (SQO), 11−7

Sequencer Load (SQL), 11−13
entering parameters, 11−13
execution times, 11−13
instruction parameters, C−12
operation, 11−14
valid addressing modes, C−12
valid file types, C−12

Sequencer Output (SQO), 11−7
entering parameters, 11−8
execution times, 11−7
instruction parameters, C−12
using, 11−10
valid addressing modes, C−12
valid file types, C−12

sinking and sourcing circuits
overview, 2−3
wiring examples, 2−3

slave/receiver communication, 13−2

spacing the controller, 1−14

specifications
analog input, A−6

analog output, A−6
general, A−3
general output, A−5
input, A−4
input filter response times, A−7
relay contact rating, A−5

SQC, Sequencer Compare, 11−7

SQL, Sequencer Load, 11−13

SQO, Sequencer Output, 11−7

SQR, Square Root, 8−11

Square Root (SQR), 8−11
execution times, 8−11
instruction parameters, C−12
updates to arithmetic status bits, 8−11
valid addressing modes, C−12
valid file types, C−12

status data file (S2:), 4−5

status file
descriptions, B−3
overview, B−1

STD, Selectable Timed Disable, 11−18

STE, Selectable Timed Enable, 11−18

STI, Selectable Timed Interrupt, 11−15
interrupt latency, 11−15

storing processor files
download, 4−7
power down, 4−8
power up, 4−8

STS, Selectable Timed Start, 11−20

SUB, Subtract, 8−5

Subroutine (SBR), 10−4
execution times, 10−4
instruction parameters, C−11
nesting subroutine files, 10−5
using, 10−6
valid addressing modes, C−11
valid file types, C−11

Subtract (SUB), 8−5
execution times, 8−5
instruction parameters, C−12
updates to arithmetic status bits, 8−5
valid addressing modes, C−12
valid file types, C−12

surge suppressors, 1−8
example, 1−9
for contactor, 1−10
for motor starters, 1−10
for relays, 1−10
recommended, 1−10

SUS, Suspend, 10−8

Suspend (SUS), 10−8
entering parameters, 10−8
execution times, 10−8
instruction parameters, C−13
valid addressing modes, C−13
valid file types, C−13

system configuration, DH-485 connection examples, D−19

system connection, 3−1

T

Temporary End (TND), 10−8
execution times, 10−8
instruction parameters, C−13
valid addressing modes, C−13
valid file types, C−13

timer file (T4:), 4−5

timer instructions
overview
addressing structure, 6−9
entering parameters, 6−8
Retentive Timer (RTO), 6−14
Timer Off-Delay (TOF), 6−12
Timer On-Delay (TON), 6−11

Timer Off-Delay (TOF), 6−12
execution times, 6−12
instruction parameters, C−13
using status bits, 6−12
valid addressing modes, C−13
valid file types, C−13

Timer On-Delay (TON), 6−11
  execution times, 6−11
  instruction parameters, C−13
  using status bits, 6−11
  valid addressing modes, C−13
  valid file types, C−13

timing diagram, message instruction, 13−8

TND, Temporary End, 10−8

TOD, Convert to BCD, 9−3

TOF, Timer Off-Delay, 6−12

TON, Timer On-Delay, 6−11

troubleshooting
  automatically clearing faults, 14−6
  contacting Allen-Bradley for assistance, P−6, 14−10
  controller error recovery model, 14−5
  determining controller faults, 14−2
  identifying controller faults, 14−6
  manually clearing faults, 14−6
  understanding the controller LED status, 14−2
  using the fault routine, 14−6

U

understanding file organization, 4−4
  addressing data files, 4−10
  numeric constants, 4−13
  processor file overview, 4−4
  specifying indexed addresses, 4−12
  specifying logical addresses, 4−10
  using the file indicator (#), 4−13

up counter
  operation, 12−8
  overview, 12−7

up counter with reset and hold
  operation, 12−8
  overview, 12−7

update times, analog inputs, 5−3

updating the high-speed counter accumulator, 12−24

user interrupt latency, B−24

V

valid addressing modes, C−1

varistors
  example, 1−9
  recommended, 1−9

voltage ranges, discrete, 2−7

W

wire types, 2−4

wiring
  analog, 2−21
  analog channels, 2−22

wiring diagrams, 2−7
  1761-L10BWA, 2−9
  1761-L10BWB, 2−12
  1761-L16AWA, 2−7
  1761-L16BBB, 2−16
  1761-L16BWA, 2−10
  1761-L16BWB, 2−13
  1761-L20AWA-5A, 2−18
  1761-L20BWA-5A, 2−19
  1761-L20BWB-5A, 2−20
  1761-L32AAA, 2−15
  1761-L32AWA, 2−8
  1761-L32BBB, 2−17
  1761-L32BWA, 2−11
  1761-L32BWB, 2−14

wiring recommendations, 2−4

X

XIC, Examine if Closed, 6−4

XIO, Examine if Open, 6−4

XOR, Exclusive Or, 9−20

Rockwell Automation
Allen-Bradley

Allen-Bradley, a Rockwell Automation Business, has been helping its customers improve productivity and quality for more than 90 years. We design, manufacture and support a broad range of automation products worldwide. They include logic processors, power and motion control devices, operator interfaces, sensors and a variety of software. Rockwell is one of the world's leading technology companies.

Worldwide representation.

Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Ireland • Israel • Italy Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • Netherlands • New Zealand • Norway • Pakistan • Peru • Philippines • Poland • Portugal Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Sweden • Switzerland • Taiwan • Thailand Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

Allen-Bradley Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1761-6.3 – July 1998
Supersedes Publication 1761-6.3 – December 1997

PN 955133-63
Copyright 1998 Rockwell International Corporation Printed in USA